

alta vista: SEARCH

[Search](#)[Live!](#)[Shopping](#)[Raging Bull](#)[Free Internet Access](#)[Email](#)

Web Results

Find this:

Example: +weather +forecast

Language: [Help](#)
[Family Filter](#)
[Language Settings](#)[Advanced Web Search](#)

Click a tab for more results on mirror packages on remote si...

[What is a tab?](#)[Products](#)[News](#)[Discussion Groups](#)[The Web](#)[Images](#)[MP3/Audio](#)[Video](#)[Directory](#)

WEB PAGES 10,418,340 pages found.

Family Filter is off

[Shop Now!](#)
[amazon.com](#)[Search: MIRROR P...](#)
[Books Music Movies](#)
[Toys Electronics](#)Search Guide: [Use the Web as a homework tool!](#)1. [Sandia National Laboratories - HTML Reference Manual \(removed\)](#)

HTML Reference Manual (removed) Neither the main page nor any of the companion sub-pages of Sandia National Laboratories' HTML Reference Manual had...

URL: www.sandia.gov/sci_compute/html_ref.html

Last modified on: 11-Jan-1999 - 5K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]2. [PHP: PHP Manual Quick Reference](#)

PHP Manual Quick Reference. Here is a list of all the PHP 3.0 functions. Click on any one of them to jump to that page in the manual. abs acos...

URL: www.php.net/quickref.php3

Last modified on: 21-Oct-1999 - 76K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)] [Facts about: [Circle Net Inc](#)]3. [No Title](#)

MIRROR(1L) MISC. REFERENCE MANUAL PAGES MIRROR(1L) NAME mirror - mirror packages on remote sites SYNOPSIS mirror [-dvTn] [-Ufilename...]

URL: nic.funet.fi/FUNET/hamster/mirror.txt

Last modified on: 2-Dec-1993 - 19K bytes - in English

[[Translate](#)] [[Related pages](#)]4. [No Title](#)

MIRROR(1L) MISC. REFERENCE MANUAL PAGES MIRROR(1L) NAME mirror - mirror packages on remote sites SYNOPSIS mirror [-dvTn] [-Ufilename...]

URL: ftp.funet.fi/funet/hamster/mirror.txt

Last modified on: 2-Dec-1993 - 19K bytes - in English

[[Translate](#)] [[Related pages](#)]5. [The Registrar Reference Manual](#)

Browse. Directories. OUR Newswire. Administrative Forms. Course Summary. ISIS Screens & Procedures. OUR Departments. Registrar Documentation....

URL: www.our.psu.edu/

Last modified on: 25-Jan-2000 - 7K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)] [Facts about: [Pennsylvania Stat...](#)]6. [Space Shuttle News Reference Manual](#)

alta vista: SEARCH

[Search](#)[Live!](#)[Shopping](#)[Raging Bull](#)[Free Internet Access](#)[Email](#)

Web Results

Find this:

Tip: Use quotes for multi-word phrases.

[Help](#)[Family Filter](#)[Language Settings](#)Language: [Advanced Web Search](#)

Click a tab for more results on ftpget+article 3893 of comp....

[What is a tab?](#)[Products](#)[News](#)[Discussion Groups](#)[The Web](#)[Images](#)[MP3/Audio](#)[Video](#)[Directory](#)

WEB PAGES 35,504 pages found.

[Family Filter is off](#)**Shop Here!**
amazon.comSearch: FTPGET+A...
[Books](#) [Music](#) [Movies](#)
[Toys](#) [Electronics](#)1. **No Title**

Article 3893 of comp.lang.perl: Xref: feenix.metronet.com comp.lang.perl:3893 Path:...

URL: www.metronet.com/perlinfo/scripts/ftpstuff/ftpget

Last modified on: 1-Jul-1993 - 22K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]2. **No Title**

Article 3893 of comp.lang.perl: Xref: feenix.metronet.com comp.lang.perl:3893 Path:...

URL: ftp.loxinfo.co.th/pub/unix/programming/perl/ftp/ftpget

Last modified on: 1-Jul-1993 - 22K bytes - in English

[[Translate](#)] [[Related pages](#)]3. **No Title**

Article 3893 of comp.lang.perl: Xref: feenix.metronet.com comp.lang.perl:3893 Path:...

URL: kawnug.oznet.ksu.edu/perl/SCRIPTS/ftpstuff/ftpget

Last modified on: 27-Mar-1996 - 22K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)] [Facts about: [Kansas State Univ...](#)]4. **Comprehensive Perl Archive Network**

CPAN: Comprehensive Perl Archive Network. Welcome to CPAN! Here you will find All Things Perl. CPAN is the Comprehensive Perl Archive Network....

URL: www.cpan.org/

Last modified on: 1-Nov-1999 - 3K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]5. **comp.lang.perl.tk FAQ**From: pvhp@lms62.lns.cornell.edu (Peter Prymmer) Newsgroups: comp.lang.perl.tk,comp.lang.perl.announce,comp.answers,news.answers Subject:...URL: w4.lns.cornell.edu/~pvhp/ptk/ptkFAQ.html

Last modified on: 27-Jul-1997 - 251K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]6. **comp.lang.perl.tk FAQ**From: pvhp@lms62.lns.cornell.edu (Peter Prymmer) Newsgroups: comp.lang.perl.tk,comp.lang.perl.announce,comp.answers,news.answers Subject:...URL: www.lns.cornell.edu/~pvhp/ptk/ptkTOC.html

Last modified on: 27-Jul-1997 - 12K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]

Web Results

Boolean query: mirror and perl

Search

Help
Family Filter
Language Settings

Tip: Include applet:[text] in your search.

About Advanced Se

Sort by:

Language: any language

☒ Show one result per Web site

From: 31/1/90

To: 31/5/94

(e.g. 31/12/99)

Main Se

WEB PAGES 105 pages found.

Family Filter is off

DVDs up to 40% off.
amazon.com

Search: MIRROR A...
Books Music Movies
Toys Electronics

11. No Title

Revision 1.30 1993/06/18 16:55:51 lmjm # Drop the uses of \$con. ## Revision 1.29 1993/06/04 15:20:34 lmjm # Patch by Joergen Haegg. to spot when no...

URL: irs.cac.psu.edu/IRS/cgi/changelogs-rev1/ftp

Last modified on: 28-Jun-1993 - 5K bytes - in English

[Translate] [More pages from this site] [Company factsheet]

12. Where can I get more information on Perl?

Where can I get more information on Perl? We'll cover five areas here: USENET (where you're probably reading this), publications, the reference...

URL: www.cis.ufl.edu/perl/faq/html/1.7.html

Last modified on: 11-Feb-1994 - 6K bytes - in English

[Translate] [More pages from this site] [Company factsheet]

13. No Title

text *- Last modified Tue Mar 30 18:10:57 1993 This is an incomplete and probably out-of-date list of all the packages distributed on...

URL: ion.apana.org.au/pub/gnu/DESCRIPTIONS

Last modified on: 3-Jan-1994 - 36K bytes - in English

[Translate]

14. Index of /lcons/Alcons/READMEs

Index of /lcons/Alcons/READMEs. <PLAINTEXT> Anthony's X Icon Library (version 1.5)

Main ReadMe File...

URL: shiva.di.uminho.pt/lcons/Alcons/READMEs

Last modified on: 18-Feb-1994 - 7K bytes - in English

[Translate]

15. No Title

usr/bin/perl # Mirror Master. # Run several mirrors in parallel. ## By Lee McLoughlin. # You can do what you like with this except claim that you...

URL: strucbio.biologie.uni-konstanz.de/pdb/mirror/mm

Last modified on: 18-Jan-1994 - 10K bytes - in English

[Translate] [More pages from this site]

16. No Title

This is the file README for the gzip distribution, version 1.2.4. gzip (GNU zip) is a compression utility designed to be a replacement for...

URL: www.centralcoastdata.org/WaterData/helpers/gzip/readme.txt

altavista: SEARCH

[Search](#) [Live!](#) [Shopping](#) [Raging Bull](#) [Free Internet Access](#) [Email](#)

Web Results

Boolean query: [Help](#) [Family Filter](#) [Language Settings](#)

Example: cat OR feline

Sort by:

Language:

From: To: (e.g. 31/12/99) [Main Se](#)

WEB PAGES 7 pages found.

[Family Filter is off](#)

Shop Now!
amazon.com

Search: HTTP.GET
Books Music Movies
Toys Electronics

1. No Title

Article 5397 of comp.lang.perl: Xref: feenix.metronet.com comp.infosystems.www:1336
comp.lang.perl:5397 Newsgroups:...
URL: ftp.telecom.sk/pub/mirror/CPAN/scripts/i...WWW/http.get.pl
Last modified on: 25-Aug-1993 - 2K bytes - in English
[\[Translate \]](#)

2. No Title

Hypertext Transfer Protocol (HTTP) Tim Berners-Lee, CERN Internet Draft Expires 5 May
1994 5 Nov 1993 Hypertext Transfer Protocol (HTTP) A Stateless...
URL: freenet.msp.mn.us/help.sav/htmldocs/http-spec.txt
Last modified on: 20-Nov-1993 - 65K bytes - in English
[\[Translate \]](#) [\[Related pages \]](#)

3. No Title

Hypertext Transfer Protocol (HTTP) Tim Berners-Lee, CERN Internet Draft Expires 5 May
1994 5 Nov 1993 Hypertext Transfer Protocol (HTTP) A Stateless...
URL: ncstrl.informatik.uni-stuttgart.de/fis/http-spec.txt
Last modified on: 31-Mar-1994 - 65K bytes - in English
[\[Translate \]](#) [\[Related pages \]](#)

4. Lynx Users Guide v2.2

Lynx Users Guide Version 2.2. Lynx is a fully-featured World Wide Web (WWW) client for users running
cursor-addressable, character-cell display...
URL: www.cs.tufts.edu/lynx/Lynx_users_guide.html
Last modified on: 16-Mar-1994 - 31K bytes - in English
[\[Translate \]](#) [\[Related pages \]](#) [\[Company factsheet \]](#)

5. No Title

local/bin/perlbin/perl ## urlget -- get a document given a WWW URL ## Jack Lund 9/3/93
zippy@ccwf.cc.utexas.edu ## from hget by: # Oscar...
URL: www.chemie.uni-dortmund.de/~loki/exp/urlget
Last modified on: 23-Mar-1994 - 4K bytes - in English
[\[Translate \]](#) [\[Related pages \]](#)

6. No Title

Hypertext Transfer Protocol (HTTP) Tim Berners-Lee, CERN Internet Draft Expires 5 May 1994 5 Nov 1993 Hypertext
Transfer Protocol (HTTP) A Stateless...

alta vista: SEARCH

[Search](#)[Live!](#)[Shopping](#)[Raging Bull](#)[Free Internet Access](#)[Email](#)

Web Results

Boolean query: mirror and perl

Search

[Help](#)[Family Filter](#)[Language Settings](#)

Example: (marketing OR sales) AND business

[Preview our NEW Advanced Search Capabilities](#)

Sort by:

Language: any language

☐ Show one result per Web site

From: 31/1/90

To: 31/3/94

(e.g. 31/12/99)

[Main Search](#)

WEB PAGES 81 pages found.

[Family Filter is off](#)Shop Here!
amazon.comSearch: MIRROR A...
[Books](#) [Music](#) [Movies](#)
[Toys](#) [Electronics](#)

1. No Title

Computer underground Digest Sun Oct 25, 1992 Volume 4 : Issue 53 Editors: Jim Thomas and Gordon Meyer (TK0JUT2@NIU.BITNET) Archivist: Brendan Kehoe...

URL: venus.soci.niu.edu/~cudigest/CUDS4/cud453.bt

Last modified on: 7-Nov-1992 - 46K bytes - in English

[\[Translate \]](#) [\[Related pages \]](#)

2. No Title

Article 5397 of comp.lang.perl: Xref: feenix.metronet.com comp.infosystems.www.1336 comp.lang.perl:5397 Newsgroups:...

URL: ftp.telecom.sk/pub/mirror/CPAN/scripts/...WWW/http.get.pl

Last modified on: 25-Aug-1993 - 2K bytes - in English

[\[Translate \]](#)

3. Perl-FAQ-1

Table of Contents. 1.1) What is Perl? 1.2) Is Perl hard to learn? 1.3) Should I program everything in Perl? 1.4) Where can I get Perl over the...

URL: ftp.sra.co.jp/public/doc/perl-faq/Perl-FAQ-1.html

Last modified on: 27-Jan-1994 - 37K bytes - in English

[\[Translate \]](#) [\[Related pages \]](#)

4. No Title

usr/bin/perl # Obey the remove commands generated but not done by mirror. # NEED TO unlink /public/micros/ibmpc/simtel20/prdsht/tc810.arc # NEED TO...

URL: www.mit.edu/afs/sipb.mit.edu/user/warlor...rrol/do_unlinks

Last modified on: 9-Aug-1993 - 316 bytes - in English

[\[Translate \]](#) [\[Related pages \]](#) [\[Company factsheet \]](#)

5. No Title

Archive-name: finding-sources Version: \$Id: csw_faq,v 1.109 1993/03/03 14:32:11 jik Exp \$ I. Table of contents This article contains the following...

URL: ftp.kiarchive.ru/pub/internet/faq/csw_faq

Last modified on: 28-Mar-1993 - 42K bytes - in English (KOI8-R)

[\[Translate \]](#) [\[Related pages \]](#)

6. How can I compare two date strings?

How can I compare two date strings? If the dates are in an easily parsed, predetermined format, then you can break

alta vista: SEARCH

[Search](#)[Live!](#)[Shopping](#)[Raging Bull](#)[Free Internet Access](#)[Email](#)

Web Results

Find this: [Help](#)[Family Filter](#)[Language Settings](#)

Tip: Use quotes for multi-word phrases.

Language: [Advanced Web Search](#)

Click a tab for more results on ftpget+article 3893 of comp....

[What is a tab?](#)[Products](#)[News](#)[Discussion Groups](#)[The Web](#)[Images](#)[MP3/Audio](#)[Video](#)[Directory](#)

WEB PAGES > 35,504 pages found.

> Family Filter is off

Shop Here!
amazon.com

Search: FTPGET+A...

[Books](#) [Music](#) [Movies](#)[Toys](#) [Electronics](#)1. No Title

Article 3893 of comp.lang.perl: Xref: feenix.metronet.com comp.lang.perl:3893 Path:...

URL: www.metronet.com/perlinfo/scripts/ftpstuff/ftpget

Last modified on: 1-Jul-1993 - 22K bytes - in English

[\[Translate \]](#) [\[More pages from this site \]](#) [\[Related pages \]](#)2. No Title

Article 3893 of comp.lang.perl: Xref: feenix.metronet.com comp.lang.perl:3893 Path:...

URL: ftp.loxinfo.co.th/pub/unix/programming/perl/ftp/ftpget

Last modified on: 1-Jul-1993 - 22K bytes - in English

[\[Translate \]](#) [\[Related pages \]](#)3. No Title

Article 3893 of comp.lang.perl: Xref: feenix.metronet.com comp.lang.perl:3893 Path:...

URL: kawng.oznet.ksu.edu/perl/SCRIPTS/ftpstuff/ftpget

Last modified on: 27-Mar-1996 - 22K bytes - in English

[\[Translate \]](#) [\[More pages from this site \]](#) [\[Related pages \]](#) [\[Facts about: Kansas State Univ...](#)4. Comprehensive Perl Archive Network

CPAN: Comprehensive Perl Archive Network. Welcome to CPAN! Here you will find All Things Perl. CPAN is the Comprehensive Perl Archive Network....

URL: www.cpan.org/

Last modified on: 1-Nov-1999 - 3K bytes - in English

[\[Translate \]](#) [\[More pages from this site \]](#) [\[Related pages \]](#)5. comp.lang.perl.tk FAQFrom: pvhp@lns62.lns.cornell.edu (Peter Prymmer) Newsgroups: comp.lang.perl.tk,comp.lang.pe
l.announce,comp.answers,news.a swers Subject:...

URL: w4.lns.cornell.edu/~pvhp/ptk/ptkFAQ.html

Last modified on: 27-Jul-1997 - 251K bytes - in English

[\[Translate \]](#) [\[More pages from this site \]](#) [\[Related pages \]](#)6. comp.lang.perl.tk FAQFrom: pvhp@lns62.lns.cornell.edu (Peter Prymmer) Newsgroups: comp.lang.perl.tk,comp.lang.pe
l.announce,comp.answers,news.a swers Subject:...

URL: www.lns.cornell.edu/~pvhp/ptk/ptkTOC.html

Last modified on: 27-Jul-1997 - 12K bytes - in English

[\[Translate \]](#) [\[More pages from this site \]](#) [\[Related pages \]](#)

Web Results

Boolean query: [Help](#)
[Family Filter](#)
[Language Settings](#)

Example (marketing OR sales) AND business

Sort by:

Language:

From: To: (e.g. 31/12/99) [Main Se](#)

[Preview our NEW Advanced Search Ce](#)

WEB PAGES 81 pages found.

[Family Filter is off](#)Shop Here!
amazon.comSearch: MIRROR A...
[Books Music Movies](#)
[Toys Electronics](#)

1. No Title

Computer underground Digest Sun Oct 25, 1992 Volume 4 : Issue 53 Editors: Jim Thomas and Gordon Meyer (TK0JUT2@NIU.BITNET) Archivist: Brendan Kehoe...

URL: venus.soci.niu.edu/~cudigest/CUDS4/cud453.txt

Last modified on: 7-Nov-1992 - 46K bytes - in English

[\[Translate \]](#) [\[Related pages \]](#)

2. No Title

Article 5397 of comp.lang.perl: Xref: feenix.metronet.com comp.infosystems.www:1336 comp.lang.perl:5397 Newsgroups:...

URL: ftp.telecom.sk/pub/mirror/CPAN/scripts/i...WWW/http.get.pl

Last modified on: 25-Aug-1993 - 2K bytes - in English

[\[Translate \]](#)

3. Perl-FAQ-1

Table of Contents. 1.1) What is Perl? 1.2) Is Perl hard to learn? 1.3) Should I program everything in Perl? 1.4) Where can I get Perl over the...

URL: ftp.sra.co.jp/public/doc/perl-faq/Perl-FAQ-1.html

Last modified on: 27-Jan-1994 - 37K bytes - in English

[\[Translate \]](#) [\[Related pages \]](#)

4. No Title

usr/bin/perl # Obey the remove commands generated but not done by mirror. # NEED TO unlink /public/micros/ibmpc/simtel20/ prdsh/tc810.arc # NEED TO...

URL: www.mit.edu/afs/sipb.mit.edu/user/warlor...rror/do_unlinks

Last modified on: 9-Aug-1993 - 316 bytes - in English

[\[Translate \]](#) [\[Related pages \]](#) [\[Company factsheet \]](#)

5. No Title

Archive-name: finding-sources Version: \$Id: csw_faq,v 1.109 1993/03/03 14:32:11 jik Exp \$ I. Table of contents This article contains the following...

URL: ftp.kiarchive.ru/pub/internet/faq/csw_faq

Last modified on: 28-Mar-1993 - 42K bytes - in English (KOI8-R)

[\[Translate \]](#) [\[Related pages \]](#)

6. How can I compare two date strings?

How can I compare two date strings? If the dates are in an easily parsed, predetermined format, then you can break

alta vista: SEARCH

[Search](#)[Live!](#)[Shopping](#)[Raging Bull](#)[Free Internet Access](#)[Email](#)

Web Results

Find this: 10383 of comp.lang.perl

[Search](#)[Help](#)[Family Filter](#)[Language Settings](#)

Example: +"natural disasters"-earthquake

Language: any language

[Advanced Web Search](#)

Click a tab for more results on 10383 of comp.lang.perl

[What is a tab?](#)[Products](#)[News](#)[Discussion Groups](#)[The Web](#)[Images](#)[MP3/Audio](#)[Video](#)[Directory](#)

WEB PAGES ▶ 19,812 pages found.

▶ Family Filter is off

Shop Now!
amazon.com

Search: 10383 OF...

[Books](#) [Music](#) [Movies](#)[Toys](#) [Electronics](#)1. No Title

bin/sh #From cs.utexas.edu!asuvax!gatech!ne s.byu.edu!effli!Warp.intel.com] ews Fri May 14:52:11 CDT 1992 #Article: 10383 of comp.lang.perl #Path:...

URL: www.metronet.com/perl/scripts/ftpstuff/ftp

Last modified on: 6-Apr-1993 - 8K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]2. Comprehensive Perl Archive Network

CPAN: Comprehensive Perl Archive Network. Welcome to CPAN! Here you will find All Things Perl. CPAN is the Comprehensive Perl Archive Network....

URL: www.cpan.org/

Last modified on: 1-Nov-1999 - 3K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]3. comp.lang.perl.tk FAQ

From: pvhp@lns62.lns.cornell.edu (Peter Prymmer) Newsgroups:

comp.lang.perl.tk,comp.lang.perl.announce,comp.answers,news.answers Subject:...

URL: w4.lns.cornell.edu/~pvhp/ptk/ptkFAQ.html

Last modified on: 27-Jul-1997 - 251K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]4. comp.lang.perl.tk FAQ

From: pvhp@lns62.lns.cornell.edu (Peter Prymmer) Newsgroups: comp.lang.perl.tk,comp.lang.perl.announce,comp.answers,news.answers Subject:...

URL: www.lns.cornell.edu/~pvhp/ptk/ptkTOC.html

Last modified on: 27-Jul-1997 - 12K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]5. PHOAKS: Frequency for comp.lang.perl.announce

People Helping One Another Know Stuff. Freq. "Together, we know it all." ;POSTER_ID=;

TARGET=_top>Pheedback. Recency. Top Posters. Area Summary....

URL: www.phoaks.com/phoaks2/newsgroups/comp/lang/perl/resources0.html

Last modified on: 3-Nov-1999 - 91K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]6. CGI Scripts and Resources

CGI scripts, desktop publishing, fonts, web authoring, and much more...

URL: desktoppublishing.com/cgi.html

Last modified on: 7-Jan-2000 - 13K bytes - in English

alta vista: SEARCH

[Search](#) [Live!](#) [Shopping](#) [Raging Bull](#) [Free Internet Access](#) [Email](#)

Web Results

Find this: 10383 of comp.lang.perl

Search

Example: + "natural disasters" - earthquake

Language: any language

[Help](#)
[Family Filter](#)
[Language Settings](#)
[Advanced Web Search](#)

Click a tab for more results on 10383 of comp.lang.perl

[What is a tab?](#)[Products](#)[News](#)[Discussion Groups](#)[The Web](#)[Images](#)[MP3/Audio](#)[Video](#)[Directory](#)

WEB PAGES 19,812 pages found.

[Family Filter is off](#)**Shop Now!**
amazon.comSearch: 10383 OF...
[Books](#) [Music](#) [Movies](#)
[Toys](#) [Electronics](#)1. No Title

bin/sh #From cs.utexas.edu!asuvax!gatech!ne s.byu.edu!eff!IWarp.intel.com[ews Fri May 14:52:11 CDT 1992 #Article: 10383 of comp.lang.perl #Path:...

URL: www.metronet.com/perl/scripts/ftpstuff/ftp

Last modified on: 6-Apr-1993 - 8K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]2. Comprehensive Perl Archive Network

CPAN: Comprehensive Perl Archive Network. Welcome to CPAN! Here you will find All Things Perl. CPAN is the Comprehensive Perl Archive Network....

URL: www.cpan.org/

Last modified on: 1-Nov-1999 - 3K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]3. comp.lang.perl.tk FAQ

From: pvhp@lns62.lns.cornell.edu (Peter Prymmer) Newsgroups: comp.lang.perl.tk,comp.lang.perl.announce,comp.answers,news.answers Subject:...

URL: w4.lns.cornell.edu/~pvhp/ptk/ptkFAQ.html

Last modified on: 27-Jul-1997 - 251K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]4. comp.lang.perl.tk FAQ

From: pvhp@lns62.lns.cornell.edu (Peter Prymmer) Newsgroups: comp.lang.perl.tk,comp.lang.perl.announce,comp.answers,news.answers Subject:...

URL: www.lns.cornell.edu/~pvhp/ptk/ptkTOC.html

Last modified on: 27-Jul-1997 - 12K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]5. PHOAKS: Frequency for comp.lang.perl.announce

People Helping One Another Know Stuff. Freq. "Together, we know it all." ;POSTER_ID=" TARGET=_top>Pheedback. Recency. Top Posters. Area Summary....

URL: www.phoaks.com/phoaks2/newsgroups/comp/lang/perl/resources0.html

Last modified on: 3-Nov-1999 - 91K bytes - in English

[[Translate](#)] [[More pages from this site](#)] [[Related pages](#)]6. CGI Scripts and Resources

CGI scripts, desktop publishing, fonts, web authoring, and much more...

URL: desktoppublishing.com/cgi.html

Last modified on: 7-Jan-2000 - 13K bytes - in English


```

L1          0 S BROADCAST? AND 4974149/PN
L2          2247 S BROADCAST? (P) SELECTIVE?
L3          2 S L2 AND 4974149/UREF
L4          70 S L2 (P) UPDAT?
L5          4 S L4 (P) HOST
L6          8 S L4 (P) REMOTE?
L7          349 S BROADCAST? (P) (SELECTIVE? (2A) (UPDAT? OR RECEIV?))
L8          26 S L7 (P) REMOTE?
L9          396 S BROADCAST? (4A) UPDAT?
L10         23 S L9 (P) SELECTIVE?

```

[illegible]

(FILE 'USPAT' ENTERED AT 13:48:26 ON 14 JUL 1999)

L1 1 S 4974149/PN
L2 0 S L1 (P) MODIF?
L3 0 S L1 (P) CHANG?
L4 241 S (ALTER? OR MODIF? OR CHANG?) (2A) DESCRIPTOR?
L5 113 S L4 (P) DATA
L6 12 S L5 (P) REQUEST?
L7 0 S L1 (P) DESCRIPTOR?
L8 1 S L1 AND ((MODIF? OR CHANG? OR CREAT?) (2A) DESCRIPTOR?)
L9 1 S L1 AND CENTRAL DIGITAL
L10 23 S USER (4A) (TRANSMIT? OR SEND? OR SENT OR PROVIDE?) (4A)
(DE
L11 45 S (SENT OR SEND? OR PROVID? OR TRANSFER? OR TRANSMIT?) (4A
) (
L12 9 S L11/AB,CLM

25 JUL 1999 13:48:26

(FILE 'USPAT' ENTERED AT 12:48:35 ON 13 JUL 1999)

L1	715 S (REQUEST? (4A) (INFORMATION OR DATA OR UPDATE)) (4A) REM
OTE	
L2	43 S L1 (P) SOFTWARE
L3	1499 S (REQUEST? (4A) (INFORMATION OR DATA OR UPDATE)) (P) ((RE
MOT	
L4	90 S L3 (P) ADDRESS? (P) SPECIF?
L5	314 S META-DATA OR METADATA
L6	0 S L1 (P) L5
L7	32 S L5 (4A) (TRANSFER? OR TRANSMIT? OR SEND? OR SENT)
L8	2 S L7 (P) REMOTE?
L9	70 S L5 (P) UPDAT?
L10	5 S L7/AB,CLM

08982157 120497

(FILE 'USPAT' ENTERED AT 08:19:55 ON 13 JUL 1999)

L1 74088 S (TRANSFER? OR SEND OR TRANSMIT?) (4A) (INSTRUCTIONS OR S
PEC
L2 194 S L1 (P) UPDAT? (P) SOFTWARE
L3 0 S L2 (P) (INTERNET OR WEB)
L4 671 S L1 (P) (INTERNET OR WEB)
L5 21 S L4 (P) UPDAT?
L6 3 S L2 (P) (CD-ROM OR CDROM)
L7 687 S L1 (P) REQUEST? (P) UPDAT?
L8 6486 S L1 (P) REMOTE
L9 1321 S L8 (P) CENTRAL?
L10 327 S (AUTOMATIC? (4A) UPDAT?) (P) (REMOTE? OR CENTRAL?)
L11 20 S L10 (P) L1
L12 11 S 5694546/UREF

267027 25728680

`_s_p_l_i_t_c_h_u_n_k` Size of chunks to split up files into
[102400]

`_s_p_l_i_t_p_a_t_t` regexp of remote pathnames to split up before
storing locally

`_l_o_c_a_l_l_s_l_R_f_i_l_e`
local file containing ls-lR - else use remote
ls_lR_file. This is useful when first mir-
roring a large package.

`_l_s_l_R_f_i_l_e` remote file containing ls-lR - else run
remote ls

`_n_a_m_e_m_a_p_p_i_n_g_s` remote to local pathname mappings (a perl `_s`
command, eg `s:old:new:`) currently only one
allowed

`_g_e_t_n_e_w_e_r` get the remote file if its date is newer than
local [true]

`_g_e_t_s_i_z_e_c_h_a_n_g_e`
get the file if size if different from local.
If a file is compressed when fetched then the
size is automatically ignored. [true]

`_c_o_m_p_r_e_s_s_p_a_t_t` regexp of files to compress before storing
locally. (See also `get_size_change`.)

`_c_o_m_p_r_e_s_s_e_x_c_l` regexp of files not to compress `[\\[zZ]]`

`_f_o_r_c_e_t_i_m_e_s` Force local times to match remote times [yes]

`_r_e_t_r_y_c_a_l_l` If initial connect fails retry ONCE after ONE
minute [yes]

`_u_p_d_a_t_e_l_o_g` Filename, relative to `local_dir`, where an
update report is to be kept

`_m_a_i_l_t_o` Mail a report to this comma separated list of
people

`_u_s_e_r` User name or uid to give to local pathnames

_g_r_o_u_p Group name or gid to give to local pathnames
 _f_i_l_e_m_o_d_e Mode to give files created locally [0444]
 _d_i_r_m_o_d_e mode to give directories created locally
 [0755]
 _t_i_m_e_o_u_t timeout ftp requests after this many seconds

Sun Release 4.1 Last change: 13 August 1991

4

MIRROR(1L) MISC. REFERENCE MANUAL PAGES MIRROR(1L)

[20]

_t_p_p_o_r_t port number of remote ftp daemon [21]
 _p_r_o_x_y set to 1 to use proxy ftp service [0]
 _p_r_o_x_y_f_t_p_p_o_r_t port number of proxy-service ftp daemon
 [4514]
 _p_r_o_x_y_g_a_t_e_w_a_y name of proxy-service, may also be supplied
 by environmental variable INTERNET_HOST
 [internet-gateway]
 _r_e_c_u_r_s_i_v_e do sub directories as well [true]
 _f_l_a_g_s_r_e_c_u_r_s_i_v_e flags to send to ls to do a recursive listing
 _f_l_a_g_s_n_o_n_r_e_c_u_r_s_i_v_e flags to send to ls to do a non-recursive
 listing
 _m_o_d_e_c_o_p_y flag indicating if we need to copy the mode
 bits [false]
 _i_n_t_e_r_a_c_t_i_v_e noninteractive copy default [false]
 _t_e_x_t_m_o_d_e transfer in binary mode by default [false]

`_f_o_r_c_e` transfer selectively by default [false]

`_g_e_t_f_i_l_e` perform get, not put by default [true]

`_v_e_r_b_o_s_e` Verbose messages [false]

`_d_i_s_c_o_n_n_e_c_t` disconnect from remote site at end of package
[false]

`*_r_e_m_o_t_e__f_s` Remote file store type. (Only copes with
unix at the moment) [unix]

`_m_a_i_l__p_r_o_g` Program called to send to the mail_to list.
[mail]

`_d_e_l_e_t_e__s_o_u_r_c_e` Delete the source files and dirs once
transferred. [false]

Each group of keywords defines how to mirror a particular package and should begin with a unique package line. The package name is used in report generation and by the -p argument, so pick something mnemonic. The minimum needed for each package is the site, remote_dir and local_dir . On

Sun Release 4.1 Last change: 13 August 1991 5

MIRROR(1L) MISC. REFERENCE MANUAL PAGES MIRROR(1L)

finding a package line all the default values are reset.

If the package name is defaults then no site is contacted but the default values given for any keywords are changed. Personally I begin my config files with:

```
package=defaults
remote_password=ukuug-soft@doc.ic.ac.uk
get_newer=yes
get_size_change=yes
```

If the package is not defaults then mirror will perform the

following steps. Unless an internal failure is detected any error will cause the current package to be skipped and the next one tried.

If Mirror is not already connected to the site it will disconnect from any site it is already connected to then attempt to connect to the remote site's ftp daemon. It will then login using the given remote username and password. Once connected mirror turns on binary mode transfers. Next it changes to the given local directory and scans it to get the details of the local files that already exist, if necessary the local directory will be created. Once this is completed the remote directory is similarly scanned. Mirror does this by changing to the remote directory and running the ftp LIST command, passing the -lRt options. (I am not very happy about this bit and hope to allow it to pull back a file containing the remote directory listing instead.) Each remote pathname will have any specified mappings performed on it to create a local pathname. Then any checks specified by the exclude_patt, max_days, get_newer and get_size_change keywords are applied on names of files or symlinks. Only exclude_patt checking is applied to directories.

The above creates a list of all required remote files and the local path names to store them in.

Once the directory listing is completed all required files are fetched from the remote site into their local path names. This is done by pulling the file into a temporary file in the target directory. If required the temporary file is compressed. The temporary file is renamed when the transfer is successful.

EXAMPLES

Here is the mirror.defaults file from the archive on src.doc.ic.ac.uk.

```
# This is the default mirror settings used by my site:
# src.doc.ic.ac.uk (146.169.3.7)
# This is home of the UKUUG Software Distribution Service
```



```
#
# Lee McLoughlin <lmjm@doc.ic.ac.uk>

# Set my defaults
package=defaults
  # Keep all local_dirs relative to here
  local_dir=/vol/public/
  remote_password=ukuug-soft@doc.ic.ac.uk
  mail_to=lmjm
  dir_mode=0755
  file_mode=0444
  user=0
  group=0
  get_newer=yes
  get_size_change=yes
  # Don't overwrite my mirror log with the remote one.
  # Don't pull back any of their mirror temporary files.
  exclude_patt=^mirror$|^MIRROR.LOG$|^in.*$|^#.*$|^lost+found/
  # Don't compress arc, zip, boo, readme files and index.txt files
  compress_excl+|.arc$|.zip$|.boo$|[Rr][Ee][Aa][Dd][Mm][Ee]|index.txt
  # Keep a log file in each updated directory
  update_log=.mirror
```

And here is part of the mirror.config:

```
#
package=gnu
  comment=Powerful and free Un*x utilities
  site=prep.ai.mit.edu
  remote_dir=/pub/gnu
  local_dir+gnu
  exclude_patt+|^ListArchives/|^lost+found/|^scheme-7.0/|^history
  # I tend to only keep the latest couple of versions of things
  # this stops mirror from re-pulling the older versions I've removed
  max_days=30
```

```
package=elisp-archive
  site=tut.cis.ohio-state.edu
  remote_dir=/pub/gnu/emacs/elisp-archive
  local_dir+gnu/EmacsBits/elisp-archive
```

```
package=X
  comment=The X Area at export
  site=export.lcs.mit.edu
  remote_dir=/contrib
  local_dir+X/contrib
  # go-1.0.b.tar.Z is immense so I store it split locally.
  exclude_patt+|^unicom|^go-1.0.b.tar.Z
```



```
# I tend to only keep the latest couple of versions of things
# this stops mirror from re-pulling the older versions I've removed
max_days=30
```

```
package=cnews
comment=The C News system
```

Sun Release 4.1 Last change: 13 August 1991

7

MIRROR(1L) MISC. REFERENCE MANUAL PAGES MIRROR(1L)

```
site=ftp.cs.toronto.edu
remote_dir=/pub/c-news
local_dir+news/c
compress_excl+|patches/PATCHDATES
compress_patt=patches/
exclude_patt+|^c-news.Z
# and on, and on ...
```

HINTS:

Always on adding in a new package check it out first turning on the `-_n` option.

If you are adding to an existing archive then it is usually best to force the timestamps so time comparisons will work.

Try and have all packages that are being retrieved from the same site one after the other. That way mirror will only have to login once.

Keep your default settings in a separate file. That way you will, hopefully, be able to share mirror details with others.

NETIQUETTE

If you are going to mirror a remote site please obey any restrictions that the site administrators place on access.

You can generally find the restrictions on connecting into the archive using the standard `ftp` command. Any restrictions are normally given as a login banner or in a, hope-

fully, obvious file.

Here are, what I hope are, some good general rules.

Only mirror a site well outside the working hours of both the local and remote sites.

It is probably unfriendly to try to mirror a remote site more than once a day.

Before trying to mirror a remote site try and find the packages you want from local archives, no one will be pleased if you soak up a lot of network bandwidth needlessly.

If you have a local archive then tell people about it so they don't have to waste bandwidth and CPU at the remote site.

Do remember to check your config files from time to time in case the remote archive has changed their access restrictions.

Sun Release 4.1 Last change: 13 August 1991 8

MIRROR(1L) MISC. REFERENCE MANUAL PAGES MIRROR(1L)

Check the remote site regularly for any new restrictions.

SEE ALSO
perl(1), ftp(1)

BUGS
The remaining keywords need to be implemented.

Should be able to mirror non Un*x sites (it may be able to but I have not tested this - the remote ls is the problem).

It should restart file transfers where they left off.

Hanging data transfers should be detected.

Article 5397 of comp.lang.perl:

Xref: feenix.metronet.com comp.infosystems.www.1336 comp.lang.perl:5397

Newsgroups: comp.infosystems.www,comp.lang.perl

Path:

feenix.metronet.com!news.ecn.bgu.edu!usenet.ins.cwru.edu!howland.reston.ans.net!xlink.net!scsing.switch.ch!news.unig
enet

From: oscar@cui.unige.ch (Oscar Nierstrasz)

Subject: perl script for connecting to http servers

Message-ID: <1993Aug27.094503.15947@news.unige.ch>

Sender: usenet@news.unige.ch

Reply-To: oscar@cui.unige.ch

Organization: University of Geneva, Switzerland

Date: Fri, 27 Aug 1993 09:45:03 GMT

Lines: 73

Well, the problems I had in my previous posting apparently didn't have to do with what I thought. (I still don't really understand what was wrong.)

Anyway, for what it is worth, here's a little perl script to grab html pages given a list of URLs. It may be useful as a start to build more elaborate scripts (like robots).

Oscar

```
#!/local/bin/perl -s
```

```
#
```

```
# hget --- get an html page from an http server
```

```
#
```

```
# Oscar Nierstrasz 26/8/93 oscar@cui.unige.ch
```

```
#
```

```
# following the example of Gene Spafford's ftpget
```

```
require "chat2.pl";
```

```
die "Usage: hget <http-url> ...\n" unless $#ARGV >= 0;
```

```
$timeout = 60;
```

```
foreach $url (@ARGV) {
```

```
    if ($url =~ m/^http://(.*)/) {
```

```
        $host = $1;
```

```
        $port = 80; # default
```

```
        $request = "/"; # default
```

```
        ($host =~ s/^[^/]+/(.*)$) && ($request = $2);
```

```
        ($host =~ s/:(\d+)/) && ($port = $1);
```

```
    }
```

```
    # relative URL, so assume previous host & port:
```

```
    elsif ($url =~ /^http:(.*)/) {
```

```
        $request = $1;
```



```

        unless ($host) {
            warn "hget: no host for $url\n";
            next;
        }
    }
    else { warn "hget: $url is not an http URL\n"; next; }
    &http_get($host,$port,$request);
}

sub http_get {
    local($host,$port,$request) = @_ ;
    ($handle = &chat'open_port($host, $port))
        || die "chat'open($host,$port): $!\n";
    &chat'print($handle,"GET $request\n")
        || die "chat'print(GET $request): $!\n";
    $done = 0;
    do{
        &chat'expect($handle, $timeout,
            '\n', q{print "$chat'thisbuf"},
            'EOF', '$done = 1',
            'TIMEOUT', 'print STDERR "TIMEOUT\n"; $done = 1'
        )
        || die "chat'expect: $!\n";
    } until $done;
    &chat'close($handle);
}

```

END

Dr. O.M. Nierstrasz, Centre Universitaire d'Informatique
 University of Geneva, 24, rue General-Dufour, CH-1211 Geneva 4, Switzerland
 Tel: +41 22 705.7664 Secr: 705.7770 Fax: 320.2927
 E-mail: oscar@cui.unige.ch Home: 733.9568

Article 3893 of comp.lang.perl:

Xref: feenix.metronet.com comp.lang.perl:3893

Path:

feenix.metronet.com!news.ecn.bgu.edu!mp.cs.niu.edu!ux1.cso.uiuc.edu!howland.reston.ans.net!gatech!asuvax!ncar!vex
per!aspen.craycos.com!scott

From: scott@craycos.com (Scott Bolte)

Newsgroups: comp.lang.perl

Subject: Re: ftpget script (provided)

Message-ID: <1993Jun30.175050.14781@craycos.com>

Date: 30 Jun 93 17:50:50 GMT

References: <C9FtMu.5zA@newsserver.technet.sg>

Organization: Cray Computer Corporation

Lines: 832

In article <C9FtMu.5zA@newsserver.technet.sg> mathias@solomon.technet.sg (Mathias Koerber) writes:

...

```
>What I'd like is a script that just goes to a specified host, logs in,
>cd's to the directory and then ftps the named files. Would be nice if it
>sat atop ftp.pl.
>
>Does anyone already have such a beast?
```

I wrote just such a beast a while ago and have been using it ever since. (In fact, it was used to pick up perl 4.036 just two nights ago.) It does not use ftp.pl, all it requires is ftp.

If you have the `at` command available it will even do the transfers at night to ease the load on the net.

Here is a brief description of the scripts shar'ed together below:

ftpget Obtains a single file from a remote host.

ftpmget	Obtains a set of files from a remote host.
----------------	--

ftpls	Run ls -l in a specific directory on a remote host.
-------	---

ftplist Given a set of hosts & directories make a series of calls to ftpls. The results are placed in files that reflect the listing source.

Enjoy,

Scott

X
X ftpvaporware A script, not yet written, that can look at the
X data provided by ftplist. It will provide
X notification that a item on an archive has
X changed. Or that a new release of a program is
X now available.
X

I would appreciate if any additions you make are sent to me. I will try
to incorporate them in any future releases.

X
X Scott Bolte (scott@craycos.com)
X 1993.06.30

END_OF_FILE

```
if test 1371 -ne `wc -c <'README'; then
    echo shar: \"README\" unpacked with wrong size!
```

```
fi
# end of 'README'
```

```
fi
if test -f 'ftpget.pl' -a "${1}" != "-c" ; then
    echo shar: Will not clobber existing file \"ftpget.pl\"
```

```
else
    echo shar: Extracting \"ftpget.pl\" \ (5666 characters\ )
    sed 's/^X//>'ftpget.pl' <<'END_OF_FILE'
```

X#!/bin/perl

X
X#####

X#
X# Copyright 1993 Scott Bolte (scott@craycos.com)

X#
X# Leave this copyright alone. But feel free to do with the script
X# as you please. Sending me enhancements would be appreciated.

X#
X# If you feel like pretending this is shareware, and want to
X# send some money my way, feel free. I promise not to object.

X#
X# Summary of ftpget:

X#
X# Obtain a file via ftp from a remote system. Anonymous ftp
X# is used. Can be asked to delay the actual ftp request.
X# On failure print out a call that can be used to try again.

X#
X# History:

X#
X# 1993.06.14 Initial implementation.
X#
X# 1993.06.15 Made remote and local file name processing safe
X# even when the file name contains white space.
X#
X# 1993.06.28 Added interface to use "at" internally.
X#

X# Examples:

X#

X# ftpget --at 23:30 prep.ai.mit.edu pub/gnu/perl-4.036.tar.gz

X#

X# ftpget --at 23:30 prep.ai.mit.edu pub/gnu/perl-4.036.tar.gz new-perl

X#

X# If you have an ftp hierarchy, as I do, the remote system can be

X# derived from the current path.

X#

X# cd ~/ftp/ftp.prep.ai.mit.edu

X# ftpget --at 23:30 - pub/gnu/perl-4.036.tar.gz

X#

X#####

X

X\$zero = \$0;

X\$zero =~ s,.*,,;

X

sub usage {

X print <<EOS;

Usage: \$zero [options] system remote_file [local_file]

X -- Stop command line processing.

X -a or --at time At the given time, which should be in hh:mm format,
X run the \$0 command.

X -d or --debug Do not run the actual ftp command, use cat instead.

X -v or --verbose Enable the verbose status message.

X -? or -h or --help Print this usage statement.

X

X If the system is "-" try to determine the remote system from
X the current path.

XEOS

X}

X

X#

X# Set the default values.

X#

X\$verbose = 0;

X\$debug = 0;

X\$cmd = "ftp -n"; # change "ftp -n" to "cat -n" for testing.

X\$time = time; # Get the time so we can recreate an "at"

X # command if need be.

X

X

X#

X# Process the command line.

X#

while (\$ARGV[0] =~ /^-./) {

X \$ _ = shift;

X if (/^--\$/) {

X last;

X }

X


```

X      if (/^-a$/ || /^--at$/ ) {
X          $delay_time = shift;
X          if ( $delay_time !~ /^d\d?:d\d$/ ) {
X              print(STDERR
X                  "Bad time specification \"$delay_time\".\n");
X                  &usage();
X                  exit(1);
X              }
X              next;
X          }
X      if (/^-d$/ || /^--debug$/ ) {
X          $debug++;
X          $cmd = "cat -n";
X          next;
X      }
X      if (/^-v$/ || /^--verbose$/ ) {
X          $verbose++;
X          next;
X      }
X
X      if (/^-\\?$/ || /^-h$/ || /^--help$/ ) {
X          &usage();
X          exit(0);
X      }
X      print "I don't recognize this switch: $_\n";
X      &usage();
X      exit(1);
X}
X
X#
X# Verify we have the right number of positional arguments.
X#
X if ( $#ARGV < 1 || $#ARGV > 2 ) {
X     &usage();
X     exit(1);
X}
X$system = $ARGV[0];
X$remote_file = $ARGV[1];
X
X#
X# If the system spec was "-" try to figure out where we are. From that
X# we might be able to construct a default host.
X#
X if ( $system eq "-" ) {
X     $system = `/bin/pwd`;
X     $system =~ s/\n//;
X     $original = $system;
X     $system =~ s,^.*\/ftp/,;;
X     $system =~ s,/.*$,;;
X     die("Could not determine system given path \"$original\".\n")

```



```

X      if $system eq "";
X      print(STDERR "Derived system is \"\$system\".\n") if $verbose;
X}
X
X#
X# Either take the optional third argument or construct it from the second.
X#
X
if ( "$remote_file" eq "" ) {
X      print(STDERR "Must specify a non-null file name.\n");
X      exit(1);
X}
if ( $#ARGV == 2 ) {
X      $local_file = $ARGV[2];
X      $explicit = 1;
X} else {
X      $local_file = $remote_file;
X      $local_file =~ s,/+$,;
X      $local_file =~ s,.*$/,;
X      $explicit = 0;
X
X      print(STDERR "Derived local path is \"\$local_file\".\n") if $verbose;
X      if ( "$local_file" eq "" ) {
X          print(STDERR "Unable to construct a local filename.\n");
X          exit(1);
X      }
X}
X
X#
X# Make the path specifications safe even when they contain spaces.
X#
X$safe_remote = $remote_file;
X$safe_local = $local_file;
X$safe_remote =~ s/(.*)/"$1"/ if $safe_remote =~ /\s/;
X$safe_local =~ s/(.*)/"$1"/ if $safe_local =~ /\s/;
X
X#
X# If a delay was asked for run the command later.
X#
if ( $delay_time ne "" ) {
X      $me = $0;
X      $me .= " --verbose" if $verbose;
X      $me .= " --debug" if $debug;
X      $me .= " $system $safe_remote";
X      $me .= " $safe_local" if $explicit;
X      $at = "at $delay_time";
X      print(STDERR "At $delay_time the following command will be run:\n");
X      print(STDERR " $me\n");
X      if ( $debug ) {

```



```

X      print(STDERR "Skipping command in debug mode.\n");
X      exit(0);
X  }
X  open(CMD, "|$at") || die("Could not run command ($at). $!\n");
X  print(CMD $me);
X  close(CMD);
X  exit(0);
X}
X
X#
X# Build the batch ftp command.
X#
X$user    = (getpwuid($<))[0]; # safe when run from "at".
X$localhost = `hostname`; chop($localhost);
if ( $localhost !~ /\. / ) {
X      #
X      # If the host name does not have '.' notation try to
X      # get an alias. We then hope it is in domain name notation.
X      #
X      @fullhost = gethostbyname($localhost);
X      $localhost = $fullhost[1] if $fullhost[1] ne "";
X}
X$template =
X"      open $system
X      user anonymous ${zero}4$user@$localhost
X      bin
X      get $safe_remote $safe_local
X      bye
X";
X
X#
X# Run the batch ftp command.
X#
Xprint(STDERR "Running command ($cmd).\n") if $verbose;
Xopen(CMD, "|$cmd") || die("Could not start command ($cmd). $!\n");
Xprint CMD $template;
Xclose(CMD);
X
X
X# Note whether or not the local file was obtained.
X#
Xif ( -f $local_file ) {
X      print("Obtained \"$local_file\" from $system.\n");
X} else {
X      $pwd=`pwd`;
X      chop($pwd);
X      @time = localtime($time);
X      $next = sprintf("%02d:%02d", $time[2], $time[1]);
X
X      $sep = "\t\\n\t ";

```



```

X      $cmd = "$0";
X      $cmd = "$sep--at $next";
X      $cmd = "$sep$system";
X      $cmd = "$sep$safe_remote";
X      $cmd = "$sep$safe_local"          if $explicit;
X
X      print("
Unable to obtain \"$local_file\" from $system.
To try again the following command might be used:
X
cd $pwd;
X$cmd
X");
X}
END_OF_FILE
if test 5666 -ne `wc -c <'ftpget.pl'`; then
    echo shar: \"'ftpget.pl'\" unpacked with wrong size!
fi
chmod +x 'ftpget.pl'
# end of 'ftpget.pl'
fi
if test -f 'ftplist.pl' -a "${1}" != "-c" ; then
    echo shar: Will not clobber existing file \"'ftplist.pl'\"
else
    echo shar Extracting \"'ftplist.pl'\" (1543 characters)
    sed "s/^X//" >'ftplist.pl' <<END_OF_FILE
X#!/bin/perl
X
X#####
X#
X# Copyright 1993 Scott Bolte (scott@craycos.com)
X#
X# Leave this copyright alone. But feel free to do with the script
X# as you please. Sending me enhancements would be appreciated.
X#
X# If you feel like pretending this is shareware, and want to
X# send some money my way, feel free. I promise not to object.
X#
X# Summary of ftplist:
X#
X# Run ftpls on a bunch of systems. The results are put in files
X# whose names map to the system/directory pair. Older copies are
X# renamed before the new edition is obtained.
X#
X# The expectation is that additional scripts will be run after
X# this one. They will compare the old and new listings to note
X# changes.
X#
X# History:
X#

```



```

X# - 1993.06.28 Initial implementation.
X#
X# Examples:
X#
X# ftplist
X#
X#####
X
X%set = (
X     "agate.berkeley.edu",          "pub/386BSD/386bsd-0.1/unofficial",
X     "bsd.coe.montana.edu",          "pub/patch-kit",
X     "hrd769.brooks.af.mil",         "pub/FAQ",
X     "prep.ai.mit.edu",              "pub/gnu",
X     );
X
foreach $system (sort(keys(%set))) {
X     $file = "$system:$set{$system}";
X     $file =~ s/,/_g;
X     $old = "$file.OLD";
X     if ( -f $file ) {
X         $error = "Could not rename \"$file\" to \"$old\". $!\n";
X         $error .= "New listing of $set{$system} will not be obtained.\n";
X         rename($file, $old) || (warn($error), next);
X     }
X     $cmd = "ftpls $system $set{$system} > $file";
X     system($cmd) && die("Could not run command ($cmd). $!\n");
X}
END_OF_FILE
if test 1543 -ne `wc -c <'ftplist.pl'`; then
    echo shar: \"'ftplist.pl'\" unpacked with wrong size!
fi
# end of 'ftplist.pl'
fi
if test -f ftpls.pl -a "${1}" != "-c" ; then
    echo shar: Will not clobber existing file \"'ftpls.pl'\"
else
    echo shar: Extracting \"'ftpls.pl'\" \$(1853 characters\))
    sed "s/^X//\" >'ftpls.pl' <<'END_OF_FILE'
X#!/bin/perl
X
X#####
X#
X# Copyright 1993 Scott Bolte (scott@craycos.com)
X#
X# Leave this copyright alone. But feel free to do with the script
X# as you please. Sending me enhancements would be appreciated.
X#
X# If you feel like pretending this is shareware, and want to
X# send some money my way, feel free. I promise not to object.
X#

```


X# Summary of ftpls:

X#

X# Do a ls in a specific directory on a remote ftp system.

X# Anonymous ftp is used.

X#

X# History:

X#

X# 1993.06.28 Initial implementation.

X#

X# Examples:

X#

X# ftpls remote_system remote_dir

X#

X#####

X

X\$cmd = "ftp -n"; # change "ftp -n" to "cat -n" for testing.

X\$zero = \$0;

X\$zero =~ s,./,;;

X

X#####

X#

X# Verify the arguments

X#

if (\$#ARGV != 1) {

X print(STDERR "Usage: \$zero system remote_dir\n");

X exit(1);

X}

X

X\$system = \$ARGV[0];

X\$remote_dir = \$ARGV[1];

X

X#####

X#

X# Build the batch ftp command.

X#

X\$user = (getpwuid(\$<))[0]; # safe when run from "at".

X\$localhost = `hostname`; chop(\$localhost);

if (\$localhost !~ /\. /) {

X #

X # If the host name does not have '.' notation try to

X # get an alias. We then hope it is in domain name notation.

X #

X @fullhost = gethostbyname(\$localhost);

X \$localhost = \$fullhost[1] if \$fullhost[1] ne "";

X}

X\$template =

X" open \$system

X user anonymous \${zero}4\$user@\$localhost

X bin

X cd \$remote_dir


```

X      ls -l
X      bye
X";
X
X#####
X#
X# Run the batch ftp command.
X#
open(CMD, "|$cmd") || die("Could not start command ($cmd). $!\n");
print CMD $template;
close(CMD);
X
exit(0);
END_OF_FILE
if test 1853 -ne `wc -c <'ftpls.pl'`; then
    echo shar: \"'ftpls.pl'\" unpacked with wrong size!
fi
chmod +x 'ftpls.pl'
# end of 'ftpls.pl'
fi
if test -f 'ftpmget.pl' -a "${1}" != "-c" ; then
    echo shar: Will not clobber existing file \"'ftpmget.pl'\"
else
    echo shar: Extracting \"'ftpmget.pl'\" (7128 characters)
    sed "s/^X//" >'ftpmget.pl' <<'END_OF_FILE'
X#!/bin/perl
X
X#####
X#
X# Copyright 1993 Scott Bolte (scott@craycos.com)
X#
X# Leave this copyright alone. But feel free to do with the script
X# as you please. Sending me enhancements would be appreciated.
X#
X# If you feel like pretending this is shareware, and want to
X# send some money my way, feel free. I promise not to object.
X#
X# Summary of ftpmget:
X#
X# Obtain a set of files via ftp from a remote system. Anonymous
X# ftp is used. Can be asked to delay the command until a later
X# time. Such a request will result in "at" being used.
X#
X# History:
X#
X# 1993.06.14 Initial implementation.
X#
X# 1993.06.15 Changed so that if a file contains white space
X# it is obtained with get instead of mget.
X#

```


X# 1993.06.30 Added --at option to allow delayed operation.

X#

X# Examples:

X#

X# ftpmget hrd769.brooks.af.mil pub/FAQ FAQ_07 FAQ_09

X#

X# cd ~/ftp/hrd769.brooks.af.mil

X# ftpmget - pub/FAQ FAQ_07 FAQ_09

X#

X# ftpmget hrd769.brooks.af.mil pub/FAQ - << EndOfList

X# FAQ_07

X# FAQ_09

X# EndOfList

X#

X#####

X

X\$zero = \$0;

X\$zero =~ s,.*/,,;

X

sub usage {

X

X print <<EOS;

Usage: \$zero [options] system remote_dir file1 [... fileN]

X or

X \$zero [options] system remote_dir -

X

X -- Stop command line processing.

X -a or --at time At the given time, which should be in hh:mm format,
X run the \$0 command.

X -d or --debug Do not run the actual ftp command, use cat instead.

X -v or --verbose Enable the verbose status message.

X -? or -h or --help Print this usage statement.

X

X If the system is "-" try to determine the remote system from
X the current path.

X

X If a "-" is given instead of a list of files the list is read
X from standard input.

XEOS

X}

X

X#####

X#

X# Set the default values.

X#

X

X\$verbose = 0;

X\$debug = 0;

X\$cmd = "ftp -n"; # change "ftp -n" to "cat -n" for testing.

X\$time = time; # Get the time so we can recreate an "at"


```

X                                     # command if need be.
X$pwd    = `/bin/pwd`; chop($pwd);
X
X#####
X#
X#      Process the command line.
X#
X
while ($ARGV[0] =~ /^-./) {
X      $_ = shift;
X      if (/^--$/) {
X          last;
X      }
X
X      if (/^-a$/ || /^--at$/ ) {
X          $delay_time = shift;
X          if ( $delay_time !~ /^d\d?:d\d$/ ) {
X              print(STDERR
X                  "Bad time specification \"$delay_time\".\n");
X                  &usage();
X                  exit(1);
X              }
X              next;
X          }
X          if (/^-d$/ || /^--debug$/ ) {
X              $debug++;
X              $cmd = "cat -n";
X              next;
X          }
X          if (/^-v$/ || /^--verbose$/ ) {
X              $verbose++;
X              next;
X          }
X          if (/^-l?$/ || /^-h$/ || /^--help$/ ) {
X              &usage();
X              exit(0);
X          }
X          print "I don't recognize this switch: $_\n";
X          &usage();
X          exit(1);
X}
X#####
X#
X# Verify the positional arguments
X#
if ( $#ARGV < 2 ) {
X    &usage();
X    exit(1);

```



```

X}
X
X$system      = $ARGV[0];
X$remote_dir  = $ARGV[1];
X
X#####
X#
X# If the system spec was "-" try to figure out where we are. From that
X# we might be able to construct a default host.
X#
if ( $system eq "-" ) {
X    $system = `/bin/pwd`;
X    $system =~ s/\n//;
X    $original = $system;
X    $system =~ s,^.*\/ftp/,;
X    $system =~ s,/.*$,;
X    die("Could not determine system given path \"$original\".\n")
X        if $system eq "";
X    print(STDERR "Derived system is \"$system\".\n") if $verbose;
X}
X
X#####
X#
X# Get the list of requested files. Either take the files from
X# the command line or from stdin.
X#
if ( $ARGV[2] ne "-" ) {
X    for( $argc = 2; $argc <= $#ARGV; $argc++) {
X        $file = $ARGV[$argc];
X        push(@files, $file);
X    }
X} else {
X    while(<STDIN>) {
X        chop;
X        $file = $_;
X        $file =~ s/^\s+//;          # leading white space
X        $file =~ s/\s+$//;          # trailing white space
X        $file =~ s/^(.*)"$1"/; # Remove enclosing quotes
X        push(@files, $file);
X    }
X}
X
X#
X# Make spaces safe for all mankind. N. Armstrong.
X#
foreach $file (@files) {
X    next if $file eq "";          # skip empty names
X    $file =~ s/(.*)"$1"/ if $file =~ /\s/; # add quotes if need be.
X    push(@tmp, $file);
X}

```



```

X@files = @tmp;
undef      @tmp;
X
X#####
X#
X#      If a delay was asked for run the command later.
X#
X
Xif ( $delay_time ne "" ) {
X    $me = $0;
X    $me .= " --verbose"          if $verbose;
X    $me .= " --debug"           if $debug;
X    $me .= " $system $remote_dir - << End_Of_List\n";
X    foreach $file (@files) {
X        $me .= "\t$file\n";
X    }
X    $me .= "End_Of_List\n";
X
X    $at = "at $delay_time";
X
X    print(STDERR "At $delay_time the following command will be run:\n");
X    print(STDERR " $me\n");
X    if ( $debug ) {
X        print(STDERR "Skipping command in debug mode.\n");
X        exit(0);
X    }
X    open(CMD, "|$at") || die("Could not run command ($at). $!\n");
X    print(CMD $me);
X    close(CMD);
X    exit(0);
X}
X#####
X#
X#      Build the batch ftp command.
X#
X
X$user      = (getpwuid($<))[0]; # safe when run from "at".
X$localhost = `hostname`;
X    chop($localhost);
Xif ( $localhost !~ /\. / ) {
X    #
X    # If the host name does not have '.' notation try to
X    # get an alias. We then hope it is in domain name notation.
X    #
X    @fullhost = gethostbyname($localhost);
X    $localhost = $fullhost[1] if $fullhost[1] ne "";
X}
X$template =
X"  open $system

```



```

X user anonymous ${zero}4$user@$localhost
X bin
X prompt
X cd $remote_dir
X";
foreach $file (@files) {
X     if ( $file =~ /\s/ ) {
X         push(@space_files, $file);
X         next;
X     }
X     if ( length($line) + length($file) + 1 > 75 ) {
X         $template .= sprintf("$line\n");
X         $line = "";
X     }
X     $line = "  mget" if $line eq "";
X     $line .= " $file";
X}
X$template .= "$line\n";
foreach $file (@space_files) {
X     $template .= "      get $file\n";
X}
X$template .= "$bye\n";
X
X#####
X#
X#      Run the batch ftp command.
X#
X
open(CMD, "|$cmd") || die("Could not start command ($cmd). $!\n");
print CMD $template;
close(CMD);
X
X#####
X#
X#      Report whether or not the files were obtained.
X#
X
print("\nReport for file transfers from $system.\n");
print(" Remote directory \"$remote_dir\".\n");
print(" Local directory \"$pwd\".\n");
foreach $file (@files) {
X     if ( -f $file ) {
X         print("  Obtained \"$file\".\n");
X         next;
X     }
X
X     if ( $file =~ /^".*"$/ ) {
X         $file =~ s/^"(.*)"$/$1/;
X         if ( -f $file ) {
X             print("  Obtained \"$file\".\n");

```



```

X      next;
X    }
X  }
X    push(@again, $file);
X}
X
X#####
X#
X#    If there was a problem getting all the files print out a command
X#    that can try again later.
X#
X
X  if ( $#again >= $[ ] {
X    @time = localtime($time);
X    $next = sprintf("%02d:%02d", $time[2], $time[1]);
X
X    print <<EOS;
X
Unable to obtained some files from $system.
To try again the following command might be used:
X
cd $pwd
X$0 --at $next $system $remote_dir - << End_Of_List
XEOS
X
X    foreach $file (@again) {
X      print("      $file\n");
X    }
X    print("End_Of_List\n");
X}
END_OF_FILE
if test 7128 -ne `wc -c <'ftpmget.pl'`; then
  echo shar: \"'ftpmget.pl\" unpacked with wrong size!
fi
chmod +x 'ftpmget.pl'
# end of 'ftpmget.pl'
fi
echo shar: End of shell archive.
exit 0
--

```

Scott Bolte scott@craycos.com +1 719 540 4186
 Cray Computer Corporation, 1110 Bayfield Drive, Colorado Springs, CO 80906
 As anyone here will tell you: I speak for myself.

*** On the Internet no one can hear you scream ***

##! /bin/sh

#From cs.utexas.edu!asuvax!gatech!news.byu.edu!effliWarp.intel.com|news Fri May 1 14:52:11 CDT 1992

#Article: 10383 of comp.lang.perl

#Path: cse.uta.edu!cs.utexas.edu!asuvax!gatech!news.byu.edu!effliWarp.intel.com|news

#From: merlyn@iWarp.intel.com (Randal L. Schwartz)

#Newsgroups: comp.lang.perl

#Subject: Re: Perl FTP Interface (Need Example) (Do I Use expect.pl?)

#Message-ID: <1992May1.152710.22905@iWarp.intel.com>

#Date: 1 May 92 15:27:10 GMT

#References: <3604@ucru2.ucr.edu> <18604@ector.cs.purdue.edu>

#Sender: news@iWarp.intel.com

#Reply-To: merlyn@iWarp.intel.com (Randal L. Schwartz)

#Organization: Stonehenge; netaccess via Intel, Beaverton, Oregon, USA

#Lines: 277

#In-Reply-To: spaf@cs.purdue.EDU (Gene Spafford)

#Nntp-Posting-Host: v.iwarp.intel.com

#

#In article <18604@ector.cs.purdue.edu>, spaf@cs (Gene Spafford) writes:

#| Well, I guess now is as good a time as any.

#|

#| I have put together a "ftp library package" that allows one to

#| construct fun little ftp programs. It works well for me -- I've built

#| a mirroring program and a couple of command-line ftp commands.

#|

#| None of this is documented (I got really busy just when I finished

#| testing this). I'll include the code for the library here, and

#| the code for my two example commands. One command lets you "ls" a

#| remote directory using ftp, and the other lets you get arbitrary

#| files, in either binary or ascii mode. I'm half done with one that

#| will let you fetch a remote tree, ala "rcp -r"

#

#Well, hey, since I have a little script that does kinda the same thing

|(the one you're "half done" with), I'll post it. Amazingly enough,

#it *also* uses chat2.pl :-).

#

#It presumes a BSD-like remote host, and fails miserably on any unusual

#forms of ftpd. Try it first to see, though.

This is a shell archive. Remove anything before this line, then unpack

it by saving it into a file and typing "sh file". To overwrite existing

files, type "sh file -c". You can also feed this as standard input via

unshar, or by typing "sh <file", e.g.. If this archive is complete, you

will see the following message at the end:

#

"End of shell archive."

Contents: ftpr

Wrapped by merlyn@iwarpv on Fri May 1 08:22:59 1992

PATH=/bin:/usr/bin:/usr/ucb ; export PATH

if test -f 'ftpr' -a "\${1}" != "-c" ; then

echo shar: Will not clobber existing file \"ftpr\"

else

echo shar: Extracting \"ftpr\" \\\(5037 characters\\)


```

sed "s/^X//" >'ftpr' <<<'END_OF_FILE'
X#!/usr/bin/perl
X
X## ftpr, last update 91/08/16
X## usage: ftpr [-a] [-d] [-t timeout] [-n] hostname topdir yes-regex except-regex
X## topdir may be whitespace-separated list of topdirs
X## yes-regex defaults to . (meaning everything)
X## except-regex defaults to '' (meaning no exceptions)
X
Xpush(@INC, '/local/merlyn/lib/perl');
X
Xrequire 'chat2.pl';
X
X$| = 1; # not much output, but we like to see it as it happens
X$timeout = 60;
X$dasha = "";
X$nflag = 0;
X$host = "localhost";
X$topdir = ".";
X$yesregex = ".";
X$noregex = "";
X$user = "anonymous";
X$pass = "merlyn@iwarrior.intel.com";
X
X{
X    last unless $ARGV[0] =~ /^-/;
X    $_ = shift;
X    $trace++, redo if /^-d/; # debug mode
X    $timeout = $1, redo if /^-t(d+)/;
X    $timeout = shift, redo if /^-t/;
X    $dasha = "-a", redo if /^-a/;
X    $nflag++, redo if /^-n/;
X    die "bad flag: $_";
X}
X
X$host = shift if @ARGV;
X$topdir = shift if @ARGV;
X$yesregex = shift if @ARGV;
X$noregex = shift if @ARGV;
X
Xdie "extra args: @ARGV" if @ARGV;
X
X($Control = &chat'open_port($host,21)) || die "open control: $!";
Xdie "expected 2dd for initial banner, got $_"
X    unless ($_ = &listen($timeout)) =~ /^2\d\d/;
X&ctalk("user $user\n");
X$_ = &listen($timeout);
Xunless (/^2\d\d/) { # might be logged in already:
X    die "expected 3dd for password query, got $_"
X        unless /^3\d\d/;

```



```

X      &ctalk("pass $pass\n");
X      die "expected 2dd for logged in, got $_"
X      unless ($_ = &clisten($timeout)) =~ /^2\d\d/;
X}
X## all set up for a conversation
X
X@list = split(/\s+/, $topdir);
Xwhile ($dir = shift list) {
X    next if $seen{$dir}++;
X    print "listing $dir\n";
X    for (&list($dir)) {
X        (warn "can't parse $_ in $dir"), next
X        unless ($tag,$file) = /^(.).*\s(\S+)\s*$/;
X        push(@list, "$dir/$file") if
X        ($tag eq 'd') && ($file !~ /\^\.\.?$/);
X        if (
X            ($tag eq '-') &&
X            ("dir/$file" =~ /$yesregex/o) &&
X            ("dir/$file" !~ /$noregex/o) &&
X            (! -e "dir/$file")
X        ) {
X            print "fetching $dir/$file...\n";
X            &get("dir/$file", "dir/$file") unless $nflag;
X        }
X    }
X}
X
X## shutdown
X&ctalk("quit\n");
X&clisten(5); # for trace
X&chat'close($Control);
Xexit(0);
X
Xsub ctalk {
X    local($text) = @_;
X    print "{$text}" if $trace;
X    &chat'print($Control,$text);
X}
X
Xsub clisten {
X    local($secs) = @_;
X    local($return,$tmp);
X    while (1) {
X        $tmp = &chat'expect($Control, $secs, '(.)r?\n', "$1\n");
X        print $tmp if $trace;
X        $return .= $tmp;
X        return $return if !length($tmp) || $tmp =~ /\^d\d\d /;
X    }
X}
X
Xsub dopen {

```



```

X      local($c);
X
X      local(@ret) = &chat'open_listen();
X      &ctalk("port "
X          join(",", @ret[0,1,2,3], int($ret[4]/256), $ret[4]%256) .
X          "\n");
X      die "expected 2dd for data open, got $_"
X          unless ($_ = &clisten($timeout)) =~ /^2\d\d/;
X      $Data = $ret[5];
X}
X
X<<'END_NOT_USED';
Xsub dtalk {
X      local($text) = @_;
X      print "{D:$text}" if $trace;
X      &chat'print($Data,$text);
X}
XEND_NOT_USED
X
Xsub dlisten {
X      local($secs,$forcereturn) = @_;
X      local($return,$tmp);
X      while (1) {
X          $tmp = &chat'expect($Data, $secs,
X              '(.|\n)+', '$&',
X              TIMEOUT, "",
X              EOF, 'undef');
X          if (defined $tmp) {
X              print "[D:$tmp]" if $trace > 1;
X              $return .= $tmp;
X              return $return unless (!$forcereturn) && (length $tmp);
X              # if timeout, return what you have
X          } else { # eof
X              return $return;
X              # maybe undef
X          }
X      }
X}
X
Xsub dclose {
X      &chat'close($Data);
X}
X
X<<'END_NOT_USED';
Xsub nlst {
X      local($dir) = @_;
X      local(@files);
X      local($_,$tmp);
X
X      &dopen();

```



```

X      &ctalk("list $dasha $dir/\n");
X      die "expected 1dd for nlist, got $_"
X          unless ($_ = &clisten($timeout)) =~ /^1\d\d/;
X      $_ = "";
X      while (1) {
X          $tmp = &dlisten($timeout);
X          last unless defined $tmp;
X          $_ = $tmp;
X      }
X      @files = sort grep(!/^\.?$/, split(/\r?\n/))
X          unless /^ls: /;
X      die "expected 2dd for nlist complete, got $_"
X          unless ($_ = &clisten($timeout)) =~ /^2\d\d/;
X      &dclose();
X      @files;
X}
XEND_NOT_USED
X
Xsub list {
X    local($dir) = @_;
X    local(@files);
X    local($_,$tmp);
X
X    &dopen();
X    &ctalk("list $dasha $dir/\n");
X    die "expected 1dd for list, got $_"
X        unless ($_ = &clisten($timeout)) =~ /^(.*\n)*1/;
X    $_ = "";
X    while (1) {
X        $tmp = &dlisten($timeout);
X        last unless defined $tmp;
X        $_ = $tmp;
X    }
X    @files = grep(/^S[rxw-]{8}/, split(/\r?\n/));
X    die "expected 2dd for list complete, got $_"
X        unless ($_ = &clisten($timeout)) =~ /^2\d\d/;
X    &dclose();
X    @files;
X}
X
Xsub get {
X    local($from, $to) = @_;
X    local($todir,*OUT);
X
X    ($todir = ".$to") =~ s#(.*)/.##$1#;
X    system "mkdir -p $todir" unless -d $todir;
X    (warn "cannot create $to.TMP: $!", return
X        unless open(OUT, ">$to.TMP");
X    select((select(OUT),$|=1)[0]);
X    &ctalk("type i\n");

```



```

X      die "expected 2dd for type i ok, got $_"
X      unless ($_ = &clisten($timeout)) =~ /^2\d\d/;
X      &dopen();
X      &ctalk("retr $from\n");
X      unless (($_ = &clisten($timeout)) =~ /^1\d\d/) {
X          warn "expected 1dd for retr, got $_";
X          close(OUT);
X          unlink("$to.TMP");
X          &dclose();
X          return;
X      }
X      {
X          $_ = &dlisten($timeout,1);
X          last unless defined $_;
X          print OUT;
X          redo;
X      }
X      close(OUT);
X      unless (($_ = &clisten($timeout)) =~ /^2\d\d/) {
X          warn "expected 2dd for retr complete, got $_";
X          close(OUT);
X          unlink("$to.TMP");
X          &dclose();
X          return;
X      }
X      &dclose();
X      rename("$to.TMP", "$to") || warn "cannot rename $to.TMP to $to: $!";
X}

```

END_OF_FILE

```

if test 5037 -ne `wc -c <'ftpr'`; then
    echo shar: \"ftpr\" unpacked with wrong size!

```

```

fi
chmod +x 'ftpr'
# end of 'ftpr'

```

```

fi
echo shar: End of shell archive.
exit 0

```

```

--
/=Randal L. Schwartz, Stonehenge Consulting Services (503)777-0095 =====\
| on contract to Intel's iWarp project, Beaverton, Oregon, USA, Sol III   |
| merlyn@iwarped.intel.com ...!any-MX-mailer-like-uunet!iwarped.intel.com!merlyn |
\=Cute Quote: "Intel: putting the 'backward' in 'backward compatible'..."=====

```


CLIPPEDIMAGE= JP403230234A
PAT-NO: JP403230234A
DOCUMENT-IDENTIFIER: JP 03230234 A
TITLE: RETRIEVING METHOD FOR DATA BASE

PUBN-DATE: October 14, 1991

INVENTOR-INFORMATION:

NAME
NISHIDA, MASATOSHI
FUJIWARA, KEIKO

ASSIGNEE-INFORMATION:

NAME	COUNTRY
OKI SOFTWARE KK	N/A
OKI ELECTRIC IND'CO LTD	N/A

APPL-NO: JP02025158
APPL-DATE: February 6, 1990

INT-CL_(IPC): G06F012/00; G06F015/40

ABSTRACT:

PURPOSE: To speed up retrieval operation by providing a terminal device with a local file and a terminal controller with an update history file.

CONSTITUTION: The terminal controller 30 which repeats data between a center 10 and the terminal device 40 is provided with the update history file 31 consisting of only data which are updated newly when a master file 11 is updated. The terminal device 40, on the other hand, is provided with the local file 41 copying the master file 11 periodically. The latest data base can, therefore, be retrieved by referring to the local file 41 and update history file 31 even unless the local file 41 and master file 1 are updated at the same time. Consequently, the data base can be retrieved more efficiently and faster.

COPYRIGHT: (C) 1991, JPO&Japio

2025 RELEASE UNDER E.O. 14176

[illegible]

;

—

•

COUNTRY
N/A

•

1

67

COPYRIGHT: (C) 1992, JPO&Japio

CLIPPEDIMAGE= JP362053085A
PAT-NO: JP362053085A
DOCUMENT-IDENTIFIER: JP 62053085 A
TITLE: VIDEO TEX TERMINAL EQUIPMENT FOR USER

PUBN-DATE: March 7, 1987

INVENTOR-INFORMATION:

NAME
EDAMOTO, MASANORI

ASSIGNEE-INFORMATION:

NAME	COUNTRY
NEC CORP	N/A

APPL-NO: JP60194303
APPL-DATE: September 2, 1985

INT-CL_(IPC):' H04N007/173; G06F013/00 ; H04M011/00

ABSTRACT:

PURPOSE: To automatically display a special picture with a prescribed time interval by providing a circuit to store a picture-retrieving state and a circuit to automatically request for a picture to the video TEX center **periodically** according to the information in the said circuit.

CONSTITUTION: A **periodic** time interval is set in a time-setting command circuit 19 through a key pad 16. The circuit 19 instructs an automatic picture-request control circuit 18 to start to **automatic picture retrieval** operation at every time after the lapsing of the set time period. after being instructed, the circuit 18 obtains a desired picture from the video TEX center in accordance with the stored information which is a storage of the retrieving process after connecting with the video TEX center in a picture retrieval storage circuit 17. Until the **periodic** time interval is released through the key pad 16, the above operation is repeated, and the user can observe the latest information of a desired picture in a displayer part 15.

COPYRIGHT: (C)1987,JPO&Japio

Network Working Group
Request for Comments: 1068

A. DeSchon
R. Braden

ISI
August 1988

Background File Transfer Program (BFTP)

Status of This Memo

This memo describes an Internet background file transfer service that is built upon the third-party transfer model of FTP. No new protocols are involved. The purpose of this memo is to stimulate discussion on new Internet service modes. Distribution of this memo is unlimited.

1. Introduction

For a variety of reasons, file transfer in the Internet has generally been implemented as an interactive or "foreground" service. That is, a user runs the appropriate local FTP user interface program as an interactive command and requests a file transfer to occur in real time. If the transfer should fail to complete for any reason, the user must reissue the transfer request. Foreground file transfer is relatively simple to implement -- no subtleties of queuing or stable storage -- and in the early days of networking it provided excellent service, because the Internet/ARPANET was lightly loaded and reasonably reliable.

More recently, the Internet has become increasingly subject to congestion and long delays, particularly during times of peak usage. In addition, as more of the world becomes interconnected, planned and unplanned outages of hosts, gateways, and networks sometimes make it difficult for users to successfully transfer files in foreground.

Performing file transfer asynchronously (i.e., in "background"), provides a solution to some of these problems, by eliminating the requirement for a human user to be directly involved at the time that a file transfer takes place. A background file transfer service

[illegible]

Background file transfer has a number of potential advantages for a user:

- o No Waiting

The user can request a large transfer and ignore it until a notification message arrives through some common channel (e.g., electronic mail).

- o End-to-end Reliability

The FTC daemon can try a transfer repeatedly until it either succeeds or fails permanently. This provides reliable end-to-end delivery of a file, in spite of the source or destination host being down or poor Internet connectivity during some time period.

- o Multiple File Delivery

In order for background file transfer to be accepted in the Internet, it may have to include some "value-added" services. One such service would be an implementation of a multiple file transfer capability for all hosts. Such a facility is suggested in RFC-959 (see the description of "NLST") and implemented in some User-FTP programs.

- o Deferred Delivery

The user may wish to defer a large transfer until an off-peak period. This may become important when parts of the Internet adopt accounting and traffic-based cost-recovery mechanisms.

There is a serious human-engineering problem with background file transfer: if the user makes a mistake in entering parameters, this mistake may not become apparent until much later. This can be the cause of severe user frustration. To avoid this problem, the user interface program ought to verify the correctness of as many of the parameters as possible when they are entered. Of course, such foreground verification of parameters is not possible if the remote host to which the parameters apply is currently unreachable.

[illegible]

DeSchon & Braden

[Page 2]

user interface language for BFTP, a description of the program organization, and sample execution scripts.

2. Background File Transfer Program

2.1 General Model

In the present BFTP design, its user interface program and its FTC daemon program must execute on the same host, which we call the BFTP control host.

Through the user interface program, a BFTP user will supply all of the parameters needed to transfer a file from source host S to destination host D, where S and D may be different from the BFTP control host. These parameters include:

- o S and D host names,
- o login names and passwords on S and D hosts, and
- o S and D file names (and optionally, directories).

The user may also specify a number of optional control parameters:

- * Source file disposition -- Copy, move (i.e., copy and delete), or simply delete the source file. The default is copy.
- * Destination file operation -- Create/Replace, append to, or create a unique destination file. The default is create/replace ("STOR").
- * FTP Parameters -- Explicitly set any of the FTP type, mode, or structure parameters at S and D hosts.
- * Multiple Transfers -- Enable "wildcard" matching to perform multiple transfers.
- * Start Time -- Set the time of day for the first attempt of the transfer. The default is "now" (i.e., make the first

16	17
18	19
20	21
22	23
24	25
26	27
28	29
30	31
32	33
34	35
36	37
38	39
40	41
42	43
44	45
46	47
48	49
50	51
52	53
54	55
56	57
58	59
60	61
62	63
64	65
66	67
68	69
70	71
72	73
74	75
76	77
78	79
80	81
82	83
84	85
86	87
88	89
90	91
92	93
94	95
96	97
98	99
100	101
102	103
104	105
106	107
108	109
110	111
112	113
114	115
116	117
118	119
120	121
122	123
124	125
126	127
128	129
130	131
132	133
134	135
136	137
138	139
140	141
142	143
144	145
146	147
148	149
150	151
152	153
154	155
156	157
158	159
160	161
162	163
164	165
166	167
168	169
170	171
172	173
174	175
176	177
178	179
180	181
182	183
184	185
186	187
188	189
190	191
192	193
194	195
196	197
198	199
200	201
202	203
204	205
206	207
208	209
210	211
212	213
214	215
216	217
218	219
220	221
222	223
224	225
226	227
228	229
230	231
232	233
234	235
236	237
238	239
240	241
242	243
244	245
246	247
248	249
250	251
252	253
254	255
256	257
258	259
260	261
262	263
264	265
266	267
268	269
270	271
272	273
274	275
276	277
278	279
280	281
282	283
284	285
286	287
288	289
290	291
292	293
294	295
296	297
298	299
300	301
302	303
304	305
306	307
308	309
310	311
312	313
314	315
316	317
318	319
320	321
322	323
324	325
326	327
328	329
330	331
332	333
334	335
336	337
338	339
340	341
342	343
344	345
346	347
348	349
350	351
352	353
354	355
356	357
358	359
360	361
362	363
364	365
366	367
368	369
370	371
372	373
374	375
376	377
378	379
380	381
382	383
384	385
386	387
388	389
390	391
392	393
394	395
396	397
398	399
400	401
402	403
404	405
406	407
408	409
410	411
412	413</

DeSchon & Braden

[Page 3]

interface program.

If the transfer should fail permanently, the FTC daemon will send a notification message to the user's mailbox. In the event of a temporary failure (e.g., a broken TCP connection), the FTC daemon will log the failure and retry the transfer after some timeout period. The retry cycles will be repeated until the transfer succeeds or until some maximum number of tries specified has been reached. In either case, a notification message will then be sent to the user's mailbox.

The user can check on the progress of the transfer by reentering the BFTP user interface program, supplying a key that was defined with the request, and displaying the current status of the request. The user may then cancel the request or leave it in the queue.

The BFTP program includes a server-Telnet module, so it can be executed as a remotely-accessible service that can be reached via a Telnet connection to the BFTP well-known port (152). This allows a user on any Internet host to perform background file transfers without running BFTP locally, but instead opening a Telnet connection to port 152 on a BFTP service host. Of course, a user can also run the local BFTP user interface program directly on any host that supports it and for which the user has login privileges.

The next section discusses how BFTP uses standard FTP servers to perform the transfers, while the following section covers the user interface of BFTP.

2.2 File Transfer Mechanics for BFTP

The BFTP makes use of the "third party" or "Server-Server" model incorporated in the Internet File Transfer Protocol [RFC-959]. Thus, the FTC daemon opens FTP control connections to the existing FTP servers on source host S and destination host D and instructs them to transfer the desired file(s) from S to D. The S and D hosts may be any two Internet hosts supporting FTP servers (but at least one of them must support the FTP "PASV" command). This approach allows the implementation of a background file transfer

[illegible][illegible]

- [illegible]

[illegible][illegible]

2025 RELEASE UNDER E.O. 14176

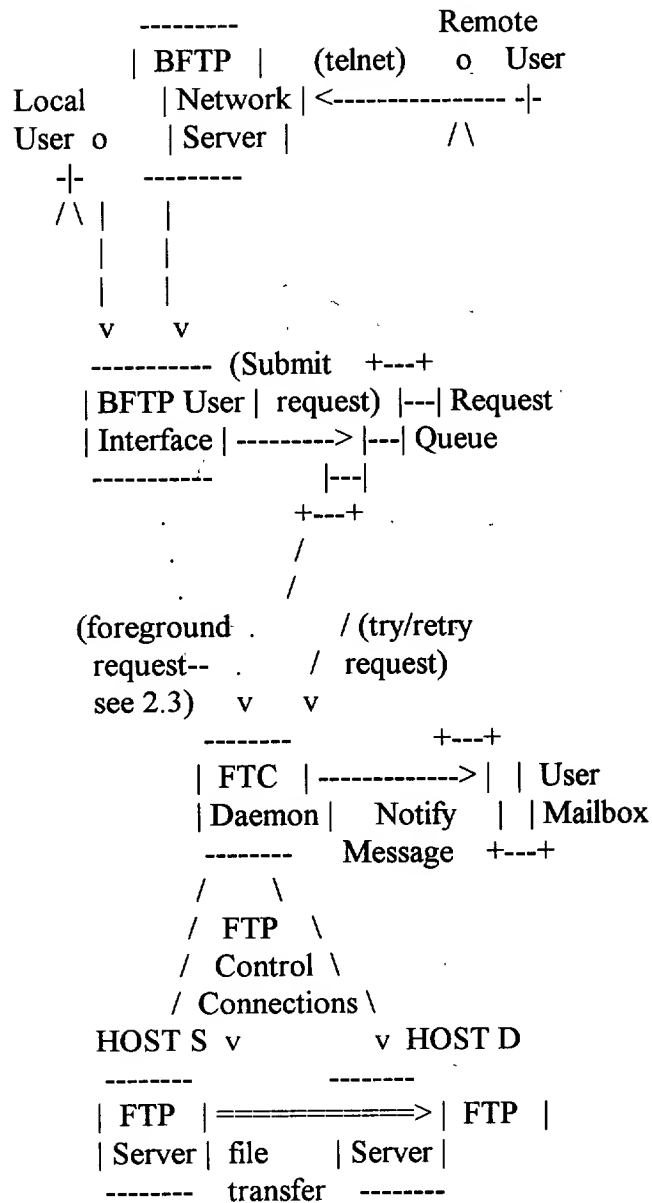
[illegible]

Figure 2 -- Server-Server File Transfer

25707 292590

Server FTP HOST S	BFTP Daemon HOST C	Server FTP HOST D
-----	-----	-----
	<----- Open TCP Ctrl conn	
	Open TCP Ctrl conn ----->	
	<----- (log in)	
(login confirm.) ----->	(log in) ----->	
	<----- (login confirm.)	
	<----- TYPE, STRU, MODE, CWD	
(confirmations) ----->	TYPE, STRU, MODE, CWD ----->	
	<----- (confirmations)	
	<----- PASV command	
PASV confirm ----->	PORT command ----->	
	<----- PORT confirm	
	RETR file ----->	
	<----- STOR file	
	<----- Open TCP Data conn	
	<----- Send file	
	<----- Close Data conn	
	<----- RETR confirm	
STOR confirm ----->		
	<----- QUIT command	
	QUIT command ----->	
Close Ctrl conn ----->		
	<----- Close Ctrl conn	

2025-07-25 14:00

BFTP currently utilizes the following Server-FTP commands [RFC-959]: USER, PASS, ACCT, PASV, PORT, RETR, STOR, STOU, CWD, NLST, MODE, STRU, TYPE, and QUIT.

The FTC daemon attempts to work around FTP servers that fail to support certain commands. For example, if a server does not support the optional command "CWD", the FTC daemon will attempt to construct a complete path name using the source directory name and the source file name. However, it is necessary that at least one of the two hosts support the FTP passive (PASV) command. While many FTP server implementations support do this command, some (in particular, the 4.2BSD FTP) do not. The PASV command was officially listed as being optional in RFC-959.

2.3 Reliable Delivery

The reliable delivery function of BFTP is analogous to reliable delivery in a transport protocol like TCP. Both depend upon repeated delivery attempts until success is achieved, and in both cases the choice of the retry interval requires some care to balance overhead against unresponsiveness.

Humans are impatient, but even their impatience has a limit. If the file cannot be transferred "soon", a human will turn to another project; typically, there is a tendency for the transfer to become less urgent the longer the wait. The FTC daemon of BFTP therefore starts each transfer request with a very short retry interval -- e.g., 10 minutes -- and then doubles this interval for successive retries, until a maximum interval -- e.g., 4 hours -- is reached. This is essentially the exponential backoff algorithm of the Ethernet, which is also used by transport protocols such as TCP, although BFTP and TCP have quite different rationales for the algorithm.

We must also define the meaning of reliable transmission for a multiple-transfer request. For example, the set of files selected by wildcard characters in a pathname is not well defined; the set may change while the request is pending, as files are created and deleted. Furthermore, it is unreasonable to regard the entire multiple transfer as a single atomic operation. Suppose that transferring a set of files fails part way through; for an atomic

1	100%
2	99%
3	98%
4	97%
5	96%
6	95%
7	94%
8	93%
9	92%
10	91%
11	90%
12	89%
13	88%
14	87%
15	86%
16	85%
17	84%
18	83%
19	82%
20	81%
21	80%
22	79%
23	78%
24	77%
25	76%
26	75%
27	74%
28	73%
29	72%
30	71%
31	70%
32	69%
33	68%
34	67%
35	66%
36	65%
37	64%
38	63%
39	62%
40	61%
41	60%
42	59%
43	58%
44	57%
45	56%
46	55%
47	54%
48	53%
49	52%
50	51%
51	50%
52	49%
53	48%
54	47%
55	46%
56	45%
57	44%
58	43%
59	42%
60	41%
61	40%
62	39%
63	38%
64	37%
65	36%
66	35%
67	34%
68	33%
69	32%
70	31%
71	30%
72	29%
73	28%
74	27%
75	26%
76	25%
77	24%
78	23%
79	22%
80	21%
81	20%
82	19%
83	18%
84	17%
85	16%
86	15%
87	14%
88	13%
89	12%
90	11%
91	10%
92	9%
93	8%
94	7%
95	6%
96	5%
97	4%
98	3%
99	2%
100	1%

BFTP addresses these issues in the following manner:

- * For a multiple file operation, the FTC daemon saves the file name list returned by the first successful NLST command in the request queue entry. This name list determines the set of source files for the transfer; there can be no later additions to the set.
- * The FTC daemon maintains a transfer status pointer. On each retry cycle, it tries to transfer only those files that have not already been successfully transferred.
- * The request is complete when all the individual file transfers have been successful, a permanent failure has occurred, or when the retry limit is reached.
- * The notification message to the user lists the status of each of the multiple files.

2.4 BFTP User Interface

The purpose of BFTP is to simplify the file transfer process and to place the burden of reliability on the BFTP control host. We have attempted to provide a "user friendly" command interface to BFTP, similar in flavor to the user interface of the TOPS-20 operating system. This interface provides extensive prompting, defaulting, and help facilities for every command.

For a list of all BFTP commands, the user may enter "?<Return>" at the main BFTP prompt ("BFTP>"). Entering "help<Return>" and "explain<Return>" will provide increasing levels of explanatory material. To obtain information on a particular command, "help <command name><Return>" may be entered. The 'quit' or 'exit' command will exit from BFTP. Command and subcommand names may be abbreviated to the shortest unique sequence for that context; alternatively, a partial name can be automatically completed by typing <Return>.

The normal procedure for a BFTP user is to set up a set of parameters defining the desired transfer and then submit the

request to the FTC daemon. To give the user the maximum flexibility, BFTP supports three modes of submission:

- o Background Operation

To request a reliable background file transfer, the user will issue the BFTP 'submit' command to the FTC daemon.

o Foreground Verification, Background Operation

The BFTP 'verify' command may be used to ascertain that file transfer parameters are valid. It causes BFTP to connect to the FTP servers on both the source and the destination hosts (if possible), log into both, verify the FTP parameters, and verify that the specified source file is present.

Once the 'verify' command has successfully completed, the user can issue the 'submit' command to schedule the actual file transfer.

o Foreground Operation

The BFTP 'transfer' command will perform the specified third-party transfer in foreground mode. This is illustrated by the dotted path bypassing the queue in Figure 1.

The easiest way to set up the parameters is to issue the 'prompt' command, which will prompt the user for all of the basic parameters required for most transfers. Certain unusual parameters must be set with the 'set' command (see Appendix B for details).

When entering any parameter, the following control characters may be used:

? will display help text for the parameter, indicating its meaning, the choices, and the default, and then reprompt for the parameter.

<ESC> will display the default value (or the last value set) for this parameter. The user can accept this default by entering <Return>, or else erase it with Control-W and enter a different value for the parameter, followed by <Return> to accept the entered value.

<Control-W>

will erase the value typed or displayed for current

parameter.

<Return>

will accept the value displayed for this parameter, and continue to the next parameter, if any. If the user has not typed a value or used <ESC> to display the default, <Return> will display the default and then accept it.

2025-10-10 10:10:10

It is important to provide a means for a user to obtain status information about an earlier request or even to cancel an earlier request. However, these functions, especially cancellation, must be controlled by some user authentication. We did not want to build a user authentication database with each BFTP instance or require login to BFTP itself, and there is no Internet-wide user authentication mechanism. We adopted the following weak authentication mechanism as a compromise:

- * When the 'submit' command is issued, it prompts the user for a character string called a "keyword", which is recorded with the request.
- * This keyword can be entered later as the argument to a 'find' command, which will display the status of all requests with matching keywords.
- * Similarly, the keyword may be used to cancel the corresponding request.

If two different users happen to choose the same keywords, of course, this scheme will not protect each other's requests from accidental or malicious cancellation. However, a notification message will be sent at the time that a cancellation occurs.

To make a series of similar requests, the user needs only to change the individual parameters that differ from the preceding request and then issue a new 'submit' command, for each request. There are commands for individually setting each of the parameters that 'prompt' sets -- and 'time' -- to provide a shortcut for BFTP experts. A simpler but lengthier procedure is to use the 'prompt' command to run through the current set of parameters, reentering the parameters that must change and using the sequence <ESC><return> to retain the previous value for each of the others. The same procedures may be used to correct a mistake made in entering a particular parameter.

The current settings of all the BFTP parameters can be displayed at any time with the 'status' command, while the 'clear' command will return all parameters to their initial values. Finally, the 'request' command allows the user to save the current set of

parameters in a file or to restore the parameters from a previously-saved file.

There is also a window-based BFTP user interface for use on a Sun Workstation, described in Appendix A. The complete list of BFTP commands is presented in Appendix B.



2025-12-19 14:02:12

3. Experience and Conclusions

BFTP has been available to users at ISI for some months. Users have reported a number of advantages of using BFTP:

- (a) Some users prefer the prompting style of BFTP to the user interface of the foreground FTP they normally use.
- (b) The BFTP "verify" command allows the user to verify that host names, passwords, and filenames are correct without having to wait for the entire transfer to take place.
- (c) Since results are returned through the mail system, a transfer can occur without tying up a terminal line, a phone line, or even a window.

BFTP must be able to communicate with a variety of Server-FTP implementations, and we have observed much variation in the commands supported, error handling, and the timing in these servers. Some of the problems we have encountered are:

- (1) Some systems (e.g., 4.2BSD) do not support the PASV command.
- (2) 4.2/3BSD systems return a non-standard response to the NLST command. Instead of returning a list of complete path-names, they use an ad hoc format consisting of a directory name followed by a list of files.
- (3) 4.2/3BSD systems may return a "permanent negative completion reply" (a 5xx FTP reply code) as a result of a communications failure such as a broken TCP connection. According to RFC-959, the appropriate response is a "transient negative completion reply" (a 4xx FTP reply code), which would inform the BFTP that the transfer should be retried.
- (4) A number of servers return badly formatted responses. An example of this is the 4.2/3BSD response to an NLST command for a non-existent file name: an error string which is not preceded by a numerical response code.

1990-1991		1991-1992		1992-1993		1993-1994		1994-1995		1995-1996		1996-1997		1997-1998		1998-1999		1999-2000		2000-2001		2001-2002		2002-2003		2003-2004		2004-2005		2005-2006		2006-2007		2007-2008		2008-2009		2009-2010		2010-2011		2011-2012		2012-2013		2013-2014		2014-2015		2015-2016		2016-2017		2017-2018		2018-2019		2019-2020		2020-2021		2021-2022		2022-2023		2023-2024		2024-2025		2025-2026		2026-2027		2027-2028		2028-2029		2029-2030		2030-2031		2031-2032		2032-2033		2033-2034		2034-2035		2035-2036		2036-2037		2037-2038		2038-2039		2039-2040		2040-2041		2041-2042		2042-2043		2043-2044		2044-2045		2045-2046		2046-2047		2047-2048		2048-2049		2049-2050		2050-2051		2051-2052		2052-2053		2053-2054		2054-2055		2055-2056		2056-2057		2057-2058		2058-2059		2059-2060		2060-2061		2061-2062		2062-2063		2063-2064		2064-2065		2065-2066		2066-2067		2067-2068		2068-2069		2069-2070		2070-2071		2071-2072		2072-2073		2073-2074		2074-2075		2075-2076		2076-2077		2077-2078		2078-2079		2079-2080		2080-2081		2081-2082		2082-2083		2083-2084		2084-2085		2085-2086		2086-2087		2087-2088		2088-2089		2089-2090		2090-2091		2091-2092		2092-2093		2093-2094		2094-2095		2095-2096		2096-2097		2097-2098		2098-2099		2099-2100		2100-2101		2101-2102		2102-2103		2103-2104		2104-2105		2105-2106		2106-2107		2107-2108		2108-2109		2109-2110		2110-2111		2111-2112		2112-2113		2113-2114		2114-2115		2115-2116		2116-2117		2117-2118		2118-2119		2119-2120		2120-2121		2121-2122		2122-2123		2123-2124		2124-2125		2125-2126		2126-2127		2127-2128		2128-2129		2129-2130		2130-2131		2131-2132		2132-2133		2133-2134		2134-2135		2135-2136		2136-2137		2137-2138		2138-2139		2139-2140		2140-2141		2141-2142		2142-2143		2143-2144		2144-2145		2145-2146		2146-2147		2147-2148		2148-2149		2149-2150		2150-2151		2151-2152		2152-2153		2153-2154		2154-2155		2155-2156		2156-2157		2157-2158		2158-2159		2159-2160		2160-2161		2161-2162		2162-2163		2163-2164		2164-2165		2165-2166		2166-2167		2167-2168		2168-2169		2169-2170		2170-2171		2171-2172		2172-2173		2173-2174		2174-2175		2175-2176		2176-2177		2177-2178		2178-2179		2179-2180		2180-2181		2181-2182		2182-2183		2183-2184		2184-2185		2185-2186		2186-2187		2187-2188		2188-2189		2189-2190		2190-2191		2191-2192		2192-2193		2193-2194		2194-2195		2195-2196		2196-2197		2197-2198		2198-2199		2199-2200		2200-2201		2201-2202		2202-2203		2203-2204		2204-2205		2205-2206		2206-2207		2207-2208		2208-2209		2209-2210		2210-2211		2211-2212		2212-2213		2213-2214		2214-2215		2215-2216		2216-2217	
-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--

The use of library routines shared between modules makes it relatively easy to implement additional user interface programs. We are currently experimenting with a window version of BFTP, the "bftptool", which runs in the SunView environment, and is described in Appendix A. Some additional interfaces that might be useful are:

- o A command line interface for use in shell scripts and "Makefiles".
- o A more general library interface which would make it easy to invoke BFTP from a variety of programs.
- o Additional full-screen form based interfaces, for example a tool running in X-Window system environment.

Lastly, BFTP would benefit from the resolution of the following open protocol issues:

- o There currently exist no provisions for Internet-wide user authentication. In the BFTP context, this means that passwords required for a file transfer must be present in BFTP request files. The security of these passwords is subject to the limitations of the file system security on the BFTP control host. Anonymous file transfer provides a partial solution, but a more general, long term solution is needed.
- o Better mechanisms are needed to cope with the diversity of real file systems in the Internet.

For example, an extension could be made to the FTP protocol to allow the daemon to learn the delimiter conventions of each host file system. This could allow a more flexible and powerful multiple-file facility in BFTP. This could include the automatic transfer of directory subtrees, for example.

4. References

[RFC-959] Postel, J., and J. Reynolds, "File Transfer Protocol (FTP)", RFC-959, USC/Information Sciences Institute,

October 1985.

DeSchon & Braden

[Page 13]

2004-2005

Appendix A -- BFTP Implementation Structure

BFTP has been implemented on both a Sun workstation running Sun OS 3.4 (based on 4.2BSD) and a VAX running 4.3BSD. The program modules are: the local user interface programs "bftp", the Internet server program "bftpd", and the FTC daemon "fts". BFTP makes use of the "at" command, a UNIX batch job facility, to submit requests and execute the daemon. An additional user interface program, the "bftp tool", is available for Sun OS 3.4, and runs in the SunView environment.

BFTP keeps its state in a set of control files: request files, command files, and message files. These files are stored in the home directory specified for the environment of the process running "bftp". If a user is running "bftp" directly, this will typically be the user's home directory. In the case where a user has made a Telnet connection to the well-known port 152 on a BFTP service host, "bftp" is started by "bftpd" (or "inetd", indirectly). As a result, the control files will be owned by the user-id under which "inetd" was started, normally "root", and stored in the top level directory "/". Note, however, that under BFTP all user files are written by the FTP servers, which are presumed to enforce the operating systems' access control conventions. Hence, BFTP does not constitute a system integrity exposure.

A.1 User Interface Program

The BFTP user interface program "bftp" may be run directly via a UNIX shell. Once the program has been started, the prompt "BFTP>" will appear and commands may be entered. These commands are described in detail in Appendix B.

A.2 Tool-Style User Interface Program

The BFTP user interface program "bftp tool" may be started from a shell window in the SunView environment on a Sun workstation. The BFTP commands may be selected via the left mouse button. The various file transfer parameters appear in a form-style interface; defaults and multiple-choice style parameter values can be filled in via menus. An advantage of this form-style interface program is that it is possible to view all of the file transfer parameters

simultaneously, providing the user with a sense for which parameter values might be mutually exclusive.

Help information can be displayed in a text subwindow by positioning the on-screen mouse pointer over a command or a parameter, and clicking the center mouse button. (No standard mechanism for displaying help information is currently included in

the SunView package.)

The commands used in the "bftptool" are for the most part very similar to the commands described in Appendix B. Request submittal and the execution of the FTC daemon are identical for the "bftp" and the "bftptool" interface programs.

A.3 Internet Server

The Internet server program "bftpd" can be invoked by opening a Telnet connection to a well-known port, and does not require login. The "bftpd" program runs under "inetd", the standard BSD4.x well-known port dispatcher. When a SYN arrives for the BFTP well-known port, "bftpd" opens the TCP connection and performs Telnet negotiations. It then passes control to the user interface "bftp" which allows the user to enter file transfer requests.

A.4 BFTP Server Daemon

The BFTP file transfer control daemon program is named "fts" (for "File Transfer Service"). This module contains code to actually cause a single file transfer operation using the FTP server-server model as shown in Figures 1 and 2. It is invoked with the command "fts <request-file>". The <request-file> contains the necessary parameters for the file transfer, in ASCII format, separated by linefeeds. Such a request file may be created by the user interface program, "bftp".

As a byproduct of the development of BFTP, "fts" represents a server-server FTP driver that can be run independent of the "bftp" program. Parameters used in the file transfer are read from a request file, which is created and accessed via library routines which can be shared between modules. This could be used to perform FTP's under program control.

2025-12-27 10:49

Appendix B: BFTP Command Summary

B.1 Special Editing Characters

In the "bftp" program, the special editing characters for command words, subcommands, and parameter fields are as follows:

<return> Accept current command/field.
<escape> Complete current command/field, or display default.
<space> Complete and delimit current command.
<delete> Erase last character.
control-L Refresh screen.
control-R Refresh line.
control-U Erase line.
control-W Erase current token.
? List legal options.

B.2 BFTP Commands

The remainder of Appendix B consists of a list of the BFTP commands. Each command should be followed by a carriage-return. In the description of the syntax for each command, square brackets "[]" are used to indicate a subcommand, or a list of possible subcommands, which are separated by the "|" character. Angle brackets "<>" are used to indicate a description of a parameter where the choices would be too numerous to list, for example "<host name/number>".

B.2.1 Clear Command

Return all parameters to their default values.

clear

B.2.2 Destination Commands

Set the destination directory.

ddir <directory name>

	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2
--	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---

8

—

[Page 16]

2021-2022

Although this may happen extremely rarely, if at all, it is possible that a system crash (or the interruption of the BFTP program) at a particularly inopportune moment may leave a request which is not queued. When the "find" command locates such a request, it displays the warning:

Your request is NOT currently queued.

If this happens, the request may be read in and resubmitted using the following procedure:

Your request is NOT currently queued.
Do you wish to change this request? yes

(BFTP displays the parameters that have been read in.)

Previous request canceled.
Use the 'submit' command to submit a new request.

B.2.5 Help Command

Print local help information.

help
help <command>

B.2.6 Quit Command

Clear parameters and exit the BFTP program.

quit

B.2.7 Prompt Command

Prompt for commonly-used parameters.

prompt

2025-09-09 10:00:00

The following are the parameters that BFTP prompts for:

copy/move/delete: [copy | move | delete]
ascii/ebcdic/image/local:
[ascii|ebcdic] [nonprint|telnet|carriage-control]

or

[image]

or

[local] <byte size>

(see "set type" for additional information)

Source --

Host: <host name/number>

User: <login>

Password: <password>

Dir: <directory including a delimiter, e.g., "/" or ">">
(either an absolute path, or relative to the login)

File: <file name>

Destination --

Host: <host name/number>

User: <login>

Password: <password>

Dir: <directory>

File: <file name>

Once the prompting has been completed, the current values of all parameters will be displayed. Parameters not mentioned in the prompting will be initialized with default values, and may be changed via the "set" commands.

[illegible]

B.2.8 Request Commands

The request commands enable the user to save a set of BFTP parameters in a "request-file" for future use. Subcommands are provided to list all available request-files, or to read, write, or delete a request-file. All request-files are stored in the user's home directory. Therefore, this facility is not available when the user is accessing BFTP by telnetting to port 152.

Delete request file "bftp-save.name".

```
request delete <name>
```

List all bftp-save files.

```
request list
```

Read a request file in as the current request.

```
request load <name>
```

Save the current request in a file named "bftp-save.name".

```
request store <name>
```

B.2.9 Set Commands

The "set" commands have complex subcommand structures and are used to set many of the less commonly used FTP parameters. The subcommands of "set" are as follows:

Set the account for the source/destination login.

```
set account [source | destination] <account string>
```

Set to true to append to destination file.

```
set append [true | false]
```

The source file will be copied to the destination file name.

100

The source file will be deleted after the file has been moved or copied.

set delete

DeSchon & Braden

[Page 20]

TABLE 1		TABLE 2		TABLE 3		TABLE 4		TABLE 5		TABLE 6		TABLE 7		TABLE 8		TABLE 9		TABLE 10		TABLE 11		TABLE 12		TABLE 13		TABLE 14		TABLE 15		TABLE 16		TABLE 17		TABLE 18		TABLE 19		TABLE 20		TABLE 21		TABLE 22		TABLE 23		TABLE 24		TABLE 25		TABLE 26		TABLE 27		TABLE 28		TABLE 29		TABLE 30		TABLE 31		TABLE 32		TABLE 33		TABLE 34		TABLE 35		TABLE 36		TABLE 37		TABLE 38		TABLE 39		TABLE 40		TABLE 41		TABLE 42		TABLE 43		TABLE 44		TABLE 45		TABLE 46		TABLE 47		TABLE 48		TABLE 49		TABLE 50		TABLE 51		TABLE 52		TABLE 53		TABLE 54		TABLE 55		TABLE 56		TABLE 57		TABLE 58		TABLE 59		TABLE 60		TABLE 61		TABLE 62		TABLE 63		TABLE 64		TABLE 65		TABLE 66		TABLE 67		TABLE 68		TABLE 69		TABLE 70		TABLE 71		TABLE 72		TABLE 73		TABLE 74		TABLE 75		TABLE 76		TABLE 77		TABLE 78		TABLE 79		TABLE 80		TABLE 81		TABLE 82		TABLE 83		TABLE 84		TABLE 85		TABLE 86		TABLE 87		TABLE 88		TABLE 89		TABLE 90		TABLE 91		TABLE 92		TABLE 93		TABLE 94		TABLE 95		TABLE 96		TABLE 97		TABLE 98		TABLE 99		TABLE 100		TABLE 101		TABLE 102		TABLE 103		TABLE 104		TABLE 105		TABLE 106		TABLE 107		TABLE 108		TABLE 109		TABLE 110		TABLE 111		TABLE 112		TABLE 113		TABLE 114		TABLE 115		TABLE 116		TABLE 117		TABLE 118		TABLE 119		TABLE 120		TABLE 121		TABLE 122		TABLE 123		TABLE 124		TABLE 125		TABLE 126		TABLE 127		TABLE 128		TABLE 129		TABLE 130		TABLE 131		TABLE 132		TABLE 133		TABLE 134		TABLE 135		TABLE 136		TABLE 137		TABLE 138		TABLE 139		TABLE 140		TABLE 141		TABLE 142		TABLE 143		TABLE 144		TABLE 145		TABLE 146		TABLE 147		TABLE 148		TABLE 149		TABLE 150		TABLE 151		TABLE 152		TABLE 153		TABLE 154		TABLE 155		TABLE 156		TABLE 157		TABLE 158		TABLE 159		TABLE 160		TABLE 161		TABLE 162		TABLE 163		TABLE 164		TABLE 165		TABLE 166		TABLE 167		TABLE 168		TABLE 169		TABLE 170		TABLE 171		TABLE 172		TABLE 173		TABLE 174		TABLE 175		TABLE 176		TABLE 177		TABLE 178		TABLE 179		TABLE 180		TABLE 181		TABLE 182		TABLE 183		TABLE 184		TABLE 185		TABLE 186		TABLE 187		TABLE 188		TABLE 189		TABLE 190		TABLE 191		TABLE 192		TABLE 193		TABLE 194		TABLE 195		TABLE 196		TABLE 197		TABLE 198		TABLE 199		TABLE 200		TABLE 201		TABLE 202		TABLE 203		TABLE 204		TABLE 205		TABLE 206		TABLE 207		TABLE 208		TABLE 209		TABLE 210		TABLE 211		TABLE 212		TABLE 213		TABLE 214		TABLE 215		TABLE 216		TABLE 217		TABLE 218		TABLE 219		TABLE 220		TABLE 221		TABLE 222		TABLE 223		TABLE 224		TABLE 225		TABLE 226		TABLE 227		TABLE 228		TABLE 229		TABLE 230		TABLE 231		TABLE 232		TABLE 233		TABLE 234		TABLE 235		TABLE 236		TABLE 237		TABLE 238		TABLE 239		TABLE 240		TABLE 241		TABLE 242		TABLE 243		TABLE 244		TABLE 245		TABLE 246		TABLE 247		TABLE 248		TABLE 249		TABLE 250		TABLE 251		TABLE 252		TABLE 253		TABLE 254		TABLE 255		TABLE 256		TABLE 257		TABLE 258		TABLE 259		TABLE 260		TABLE 261		TABLE 262		TABLE 263		TABLE 264		TABLE 265		TABLE 266		TABLE 267		TABLE 268		TABLE 269		TABLE 270		TABLE 271		TABLE 272		TABLE 273		TABLE 274		TABLE 275		TABLE 276		TABLE 277		TABLE 278		TABLE 279		TABLE 280		TABLE 281		TABLE 282		TABLE 283		TABLE 284		TABLE 285		TABLE 286		TABLE 287		TABLE 288		TABLE 289		TABLE 290		TABLE 291		TABLE 292		TABLE 293		TABLE 294		TABLE 295		TABLE 296		TABLE 297		TABLE 298		TABLE 299		TABLE	
---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	---------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-----------	--	-------	--

Set the mailbox to which the results will be returned. The mailbox should be in standard internet format, for example: "deschon@isi.edu".

set mailbox <mailbox string>

 \angle

set mode [stream | block | compress]

The source file will be deleted after it has been copied.

set move

[illegible]

set multiple [true | false]

Set the port for the source/destination FTP connection.

```
set port [source | destination] <port number>
```

Set the FTP structure.

```
set structure [file | record | page]
```

Set the FTP type and format / byte size parameters. Note that a normal text file is usually "ascii", and a "binary" file is often the same as an "image" file.

set type [ascii|ebcdic] [nonprint|telnet|carriage-control]

or

```
set type [image]
```

or

```
set type [local] <byte size>
```

Set to true if the STOU command is to be used. If the STOU command is supported by the destination host, the file will be stored into a file having a unique file name.

```
set unique [true | false]
```

from different host

Set to true to display full FTP conversations for "verify" and "transfer" commands.

set verbose [true | false]

2025-07-24 10:43:43

B.2.10 Source Commands

Set the source directory.

sdir <directory name>

Set the source file name.

sfile <file name>

Set the source host, user, and password.

shost <host name/number> <login> <password>

B.2.11 Status Command

Display the current parameter values.

status

B.2.12 Submit Command

Submit the current request for background FTP.

submit

BFTP prompts for the following information:

StartTime: <date and/or time>

ReturnMailbox: <internet mailbox>

RequestKeyword: <made-up keyword>

B.2.13 Time Command

Set the start time, the starting retry interval, and the maximum number of tries.

time <date and/or time> <minutes between tries>
<maximum number of tries>

B.2.14 Transfer Command

0000157.10049
25T03T.25T03T.25T03T

Perform the current request in the foreground.

transfer

2025-07-26 14:00:00

B.2.15 Verify Command

Make the connections now to check parameters.

verify

NOT RECORDED

Appendix C Example BFTP User Script

deschon.isi.edu 1% telnet hobgoblin.isi.edu 152
Trying 128.9.0.42 ...
Connected to hobgoblin.isi.edu.
Escape character is '^['

BFTP Server (hobgoblin.isi.edu)

Background File Transfer: For help, type '?', 'help', or 'explain'.

BFTP> prompt

Copy/Move/Delete: copy

Source --

Host: deschon.isi.edu
User: deschon
Password:
Dir: ./
File: foo*

Destination --

Host: venera.isi.edu
User: deschon
Password:
Dir: ./temp/
File: foo*

StartTime: Tue Oct 6 10:14:43 1987 (interval) 60 (tries) 5

ReturnMailbox: deschon@isi.edu

RequestPassword:

< BFTP> set multiple true

BFTP> status

Request type: COPY

Source --

Host 'deschon.isi.edu'
User 'deschon'
Pass SET

get multiple files

need to define the host again

Acct "
Dir '/'
File 'foo*'
Port 21

Destination --
Host 'venera.isi.edu'

DeSchon & Braden

[Page 24]

25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

RFC 1068

August 1988

User 'deschon'

Pass: SET

Acct "

Dir. '/temp/'

File 'foo*'

Port: 21

Structure: file, Mode: stream, Type: ascii, Format: nonprint

Multiple matching: TRUE

Return mailbox: 'deschon@isi.edu', Password: SET

Remaining tries: 5, Retry interval: 60 minutes

Start after Tue Oct 6 10:14:43 1987.

BFTP> submit

Checking parameters.

Request bftp560538880 submitted to run at 10:14 Oct 6.

BFTP> quit

bye

Connection closed by foreign host.

deschon.isi.edu 2%

deschon@isi.edu

2007-10-17 14:00

Appendix D Sample BFTP Notification Message

Received-Date: Tue, 6 Oct 87 10:15:52 PDT
Date: Tue, 6 Oct 87 10:15:47 PDT
From: root (Operator)
Posted-Date Tue, 6 Oct 87 10:15:47 PDT
To: deschon
Subject: BFTP Results: bftp560538880

Request blip560538880 submitted to run at 10:14 Oct 6.

Tue Oct 6 10 15:22 1987: starting...

Request type: COPY
Source: deschon.isi.edu-deschon-XXX--21-./-foo*
Destination: venera.isi.edu-deschon-XXX--21-./temp/-
Stru: F, **Mode:** S, **Type:** A N, **Creation:** STOR
Multiple matching: TRUE
Return mailbox: 'deschon@isi.edu', **Password:** SET
Remaining tries: 5, **Retry interval:** 60 minutes

```

Connect to deschon.isi.edu, 21
deschon.isi.edu ==> 220 deschon.isi.edu FTP server (Version 4.7
Sun Sep 14 12:44:57 PDT 1986) ready.
Connect to venera.isi.edu, 21
venera.isi.edu ==> 220 venera.isi.edu FTP server (Version 4.107
Thu Mar 19 20:54:37 PST 1987) ready.
deschon.isi.edu <== USER deschon
deschon.isi.edu ==> 331 Password required for deschon.
deschon.isi.edu <== PASS XXX
deschon.isi.edu ==> 230 User deschon logged in.
venera.isi.edu <== USER deschon
venera.isi.edu ==> 331 Password required for deschon.
venera.isi.edu <== PASS XXX
venera.isi.edu ==> 230 User deschon logged in.
deschon.isi.edu <== CWD ./
deschon.isi.edu ==> 200 CWD command okay.
venera.isi.edu <== CWD ./temp/
venera.isi.edu ==> 250 CWD command successful.
deschon.isi.edu <== PORT 128,9,1,56,4,106
deschon.isi.edu ==> 200 PORT command okay.

```


deschon.isi.edu <== NLST foo*
deschon.isi.edu ==> 150 Opening data connection for /bin/ls
(128.9.1.56,1130) (0 bytes).
deschon.isi.edu ==> 226 Transfer complete.
deschon.isi.edu <== PASV
deschon.isi.edu ==> 502 PASV command not implemented.
venera.isi.edu <== PASV

venera.isi.edu ==> 227 Entering Passive Mode (128,9,0,32,6,200)
deschon.isi.edu <== PORT 128,9,0,32,6,200
deschon.isi.edu ==> 200 PORT command okay.
deschon.isi.edu <== RETR foo
venera.isi.edu <== STOR foo
deschon.isi.edu ==> 150 Opening data connection for foo
 (128.9.0.32,1736) (0 bytes).
deschon.isi.edu ==> 226 Transfer complete.
venera.isi.edu ==> 150 Opening data connection for foo
 (128.9.1.56,20).
venera.isi.edu ==> 226 Transfer complete.
venera.isi.edu <== PASV
venera.isi.edu ==> 227 Entering Passive Mode (128,9,0,32,6,201)
deschon.isi.edu <== PORT 128,9,0,32,6,201
deschon.isi.edu ==> 200 PORT command okay.
deschon.isi.edu <== RETR foo1
venera.isi.edu <== STOR foo1
deschon.isi.edu ==> 150 Opening data connection for foo1
 (128.9.0.32,1737) (4 bytes).
deschon.isi.edu ==> 226 Transfer complete.
venera.isi.edu ==> 150 Opening data connection for foo1
 (128.9.1.56,20).
venera.isi.edu ==> 226 Transfer complete.
deschon.isi.edu <== QUIT
venera.isi.edu <== QUIT

→ connect
again to the
host for the
second file

Tue Oct 6 10 15:39 1987: completed successfully.

DeSchon & Braden

[Page 27]

Figure 1. Schematic representation of the experimental design. The figure is divided into two main sections: 'Pre-treatment' and 'Treatment'. The 'Pre-treatment' section shows a flowchart starting with 'Pre-treatment' leading to 'Baseline' (with 'Pre-treatment' and 'Baseline' labels), then to 'Pre-treatment' (with 'Pre-treatment' and 'Baseline' labels), and finally to 'Baseline' (with 'Pre-treatment' and 'Baseline' labels). The 'Treatment' section shows a flowchart starting with 'Treatment' leading to 'Baseline' (with 'Pre-treatment' and 'Baseline' labels), then to 'Pre-treatment' (with 'Pre-treatment' and 'Baseline' labels), and finally to 'Baseline' (with 'Pre-treatment' and 'Baseline' labels). The flowchart is divided into two main sections: 'Pre-treatment' and 'Treatment'. The 'Pre-treatment' section shows a flowchart starting with 'Pre-treatment' leading to 'Baseline' (with 'Pre-treatment' and 'Baseline' labels), then to 'Pre-treatment' (with 'Pre-treatment' and 'Baseline' labels), and finally to 'Baseline' (with 'Pre-treatment' and 'Baseline' labels). The 'Treatment' section shows a flowchart starting with 'Treatment' leading to 'Baseline' (with 'Pre-treatment' and 'Baseline' labels), then to 'Pre-treatment' (with 'Pre-treatment' and 'Baseline' labels), and finally to 'Baseline' (with 'Pre-treatment' and 'Baseline' labels).